Oracle® Big Data Discovery Cloud Service

Getting Started Guide

E65362-05

November 2016

Oracle Big Data Discovery Cloud Service Getting Started Guide,

E65362-05

Contents

Preface

Oracle Big Data Discovery is a set of end-to-end visual analytic capabilities that leverage the power of Apache Spark to turn raw data into business insight in minutes, without the need to learn specialist big data tools or rely only on highly skilled resources. The visual user interface empowers business analysts to find, explore, transform, blend and analyze big data, and then easily share results.

About this guide

This guide introduces Oracle Big Data Discovery, and orients you how to use it for your needs, from loading data to exploring, transforming, and updating it.

Along the way, the guide introduces key terms, components and user interfaces in Big Data Discovery. This guide is complementary to the Getting Started video series.

Audience

This guide is for business analysts, data scientists, and data engineers who work with big data.

Conventions

The following conventions are used in this document.

Typographic conventions

The following table describes the typographic conventions used in this document.

Typeface	Meaning
User Interface Elements	This formatting is used for graphical user interface elements such as pages, dialog boxes, buttons, and fields.
`Code Sample`	This formatting is used for sample code segments within a paragraph.
Variable	This formatting is used for variable values. For variables within a code sample, the formatting is ***`Variable`***.
`File Path`	This formatting is used for file names and paths.

Symbol conventions

The following table describes symbol conventions used in this document.

Symbol	Description	Example	Meaning
>	The right angle bracket, or greater-than sign, indicates menu item selections in a graphic user interface.	File > New > Project	From the File menu, choose New, then from the New submenu, choose Project.

Path variable conventions

This table describes the path variable conventions used in this document.

Path variable	Meaning
`$ORACLE_HOME`	Indicates the absolute path to your Oracle Middleware home directory, where BDD and WebLogic Server are installed.
`$BDD_HOME`	Indicates the absolute path to your Oracle Big Data Discovery home directory, `$ORACLE_HOME/BDD-<version>`.
`$DOMAIN_HOME`	Indicates the absolute path to your WebLogic domain home directory. For example, if your domain is named `bdd-<version>_domain`, then `$DOMAIN_HOME` is `$ORACLE_HOME/user_projects/domains/bdd-<version>_domain`.
`$DGRAPH_HOME`	Indicates the absolute path to your Dgraph home directory, `$BDD_HOME/dgraph`.

Contacting Oracle Customer Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. This includes important information regarding Oracle software, implementation questions, product and solution help, as well as overall news and updates from Oracle.

You can contact Oracle Customer Support through Oracle's Support portal, My Oracle Support at https://support.oracle.com.

1

Getting a Big Data Discovery Cloud Service Subscription

To subscribe to the service, check these prerequisites and contact your Oracle sales representative to obtain a subscription and activate it. Next, create a service instance, and log into Studio to explore and analyze data.

Requesting and activating a subscription
: To request a subscription to Big Data Discovery Cloud Service, contact your Oracle sales representative and provide the name and email of an account administrator for your service.

Logging into the service console for the first time
: After you receive a "Welcome" email about a BDDCS instance enabled for you, log into the Service Console URL to provision your instance.

Creating a service instance
: To create a Big Data Discovery Cloud Service instance, use the three steps in the **Create Instance** wizard: select a cluster for hosting the instance, provide configuration details and user credentials for all required underlying services, and confirm your instance configuration.

Replacing SSL certificates
: Your BDDCS instance is provisioned in the Oracle Cloud with SSL enabled in WebLogic Server by default. Enabling SSL for BDDCS activates WebLogic Server's default Demo Identity and Demo Trust Keystores. As their names suggest, these keystores are untrusted and meant for demo purposes only. After creating a BDDCS instance and running the provisioning wizard, you should replace them with your own certificates.

Verifying the service instance
: To verify all BDDCS components are running successfully, log into Studio and add a data set by uploading a CSV file. Observe it in the Studio's Catalog and **Explore** pages.

Accessing documentation for the service
: Find BDDCS documentation in the Oracle Help Center, under the Cloud tile. Use BDDCS Getting Started Guide (this document) to learn how to provision and use BDDCS.

About Big Data Discovery Cloud Service

Oracle Big Data Discovery Cloud service (BDDCS) is a single-tenant Platform as a Service (PaaS) offering in Oracle Public Cloud (OPC).

Here is how BDDCS relates to BDCS:

- **Oracle Big Data Cloud Service (BDCS)** is a "container" for the BDDCS instance. BDCS is a platform for running workloads on Hadoop systems, and for the development of Big Data applications. A BDCS starter pack consists of six hosted nodes. You can add more nodes if needed. Each BDCS instance includes access to the Cloudera Hadoop Distribution (CDH) Enterprise Data Hub Edition, and Oracle Big Data Connectors.

- **Oracle Big Data Discovery Cloud Service (BDDCS)** is deployed into an existing BDCS cluster instance in the same Identity Domain. It occupies one node in the BDCS cluster instance. BDDCS provides catalog, exploration, and visualization of data stored in a BDCS cluster.

  For an overview of tasks you can do, and for information on components in BDDCS, including Studio, see Welcome to Big Data Discovery.

Before you begin with your subscription

Before you create your first Big Data Discovery Cloud Service instance, you should have an Oracle Cloud Services account, a subscription to Oracle Big Data Cloud Service (BDCS), to Oracle Storage Cloud Service, and to VPN.

Here is a description of each of these requirements:

- **Oracle Cloud Services**. It gives you a non-metered subscription to Oracle Cloud Services (MyServices), which includes an account at Oracle.com. See Buying a Nonmetered Subscription to an Oracle Cloud Service.

- **Oracle Big Data Cloud Service (BDCS)**. It provides the BDCS cluster with Hadoop in the cloud for your BDDCS instance. The instance of BDDCS is hosted on this Hadoop cluster and runs on one of the nodes. For example, if you use a starter pack of six nodes for the Big Data Cloud Service, when you provision Big Data Discovery Cloud Service, it is created for you by Oracle Cloud administrators on one of these nodes. The BDCS version should be higher than 4.3.1. When you provision BDDCS, Oracle Cloud checks for this requirement.

- **Oracle VPN Service**. To log into BDDCS Studio, you must first establish a VPN connection to the BDCS cluster and the BDDCS instance hosted in it. If you already have a VPN connection to BDCS, you can reuse it. See Enabling VPN access to the service.

- **Oracle Storage Cloud Service**. This service stores backup files for BDDCS.

BDDCS requires that Big Data Cloud Service and Storage Cloud Service are provisioned in the same Identity Domain and under the same user account where BDDCS is provisioned.

Interfaces to the service

Use the Service console for administration tasks; Studio for exploring, transforming, and discovering data; Data Processing CLI for loading and updating data; SSH terminal for accessing the Big Data Discovery Cloud Service node in the Big Data Cloud Service cluster.

Here is a description of each of these interfaces:

- The **BDDCS console** is the main access point for you to start, stop, check the status, and back up your service instance. You see the console after you provision an instance. See Instance Life Cycle Tasks.

- The **BDDCS Studio** is the front-end application Big Data Discovery for exploring, transforming, and discovering data stored in a Big Data Cloud Service cluster. You access Studio from the BDDCS console. See Accessing Studio.

- The **Data Processing CLI** (DP CLI) is the command-line tool for automated loading of new data and updating existing data in BDDCS. You run this tool directly on the node hosting your service instance in the Big Data Cloud Service cluster. You can also run this tool programmatically and include it in your scripts. See About Data Processing.

- The **SSH terminal** lets you access the node hosting your service instance. See Accessing the service node in a cluster.

What's new in the service?

This topic describes the new features in the Big Data Discovery Cloud Service.

- The **Maintenance** tab of the **Administration** page for the service console is ready and currently empty. It will list future patches. To apply patches for moving to this release, see Patching the service.
- Validation for Studio's Administrator password is improved. The password must be between 8 and 30 characters, with at least one lower case letter, one upper case letter, one number and one special character (_, -, !, @, #, $, ^, (), [], {}, :, +, ~), and cannot be an ordinary word.
- The logic for handling error messages has been improved. The software now recognizes these categories of error messages: general information, general error, and operational error. General information messages are cleared once you refresh the page, or perform any actions in the console. General errors are cleared when you refresh the page, or go to another page in the console. You can also close the general error messages. Operational error messages persist at the top of the page until you start a new operation, such as start, stop, or back up, or until the new operation succeeds.
- You can update Hadoop credentials in the Big Data Discovery Cloud Service console. This is useful when you have previously changed them in Cloudera Manager. In the previous release, you could only update Studio and WebLogic Server credentials in the service console.
- If the Big Data Cloud Service cluster hosting the Big Data Discovery Cloud Service instance has been expanded to include more nodes, this is reflected in the number of nodes shown in **Overview** on the service's console. No user interaction is required.

In addition, for information on what is new in Big Data Discovery (on-premises implementation), see New and updated features.

Relationship between the service and Big Data Discovery

Big Data Discovery Cloud Service (BDDCS) is an implementation of Big Data Discovery in the Oracle Cloud, whereas Big Data Discovery (BDD) is an on-premise implementation.

In BDDCS, you can use BDD as if you were to install Big Data Discovery on premise at your site. Here is the summary of a few differences between BDDCS and BDD on-premise implementation:

- A cluster of BDD nodes is not supported in BDDCS. The BDD on-premise implementation can be installed as a cluster, on more than one node in the Hadoop environment. Big Data Discovery Cloud Service (BDDCS) is hosted on a single BDCS node in your Big Data Cloud Service (BDCS) implementation in the Oracle Public Cloud.
- The Enterprise Manager plug-in and the `bdd-admin` utility are not supported in BDDCS. To monitor the status of the BDDCS instance, start, stop, and back up it, use the BDDCS console in Oracle Cloud.

- Tuning of the Dgraph and HDFS flags and parameters is not supported in BDDCS. BDDCS is deployed with the Dgraph settings that let you use it out of the box, without having to tune Dgraph parameters, flags, location of logs, or log levels. You have full access to the logs for the Dgraph, Studio, Data Processing and HDFS Agent. To see any of the BDDCS component logs, access the BDDCS node in the BDCS cluster.
- In Studio, these differences exist between the Studio component in BDDCS and Studio in BDD:
  - You can use a wide variety of Studio's data visualization and analysis components, and create your own visualizations with Studio's Custom Visualization Component extension, but Studio Component SDK for customization of existing Studio components is not supported in BDDCS.
  - You can use all transformations available in Studio, and create your own transformation scripts in Studio's **Transform** editor, but extending the transform function library in Studio with custom, external Groovy scripts (to obtain additional "built-in" transform functions in Studio) is not supported in BDDCS.
  - You can view all of the Studio settings in the Control Panel, but editing some of them is not supported in BDDCS. The *BDDCS Administrator's Guide* indicates those settings whose editing is allowed in BDDCS.
  - In BDDCS, you cannot send emails of Studio's snapshot or bookmarks galleries. Also, email configuration in Studio's Control panel is disabled in BDDCS. Instead, to share Studio's snapshots and bookmarks with others, create additional Studio users who can view the same snapshots in Studio.

About users and roles

Oracle Big Data Discovery Cloud Service depends on several underlying services in Oracle Public Cloud and supports the following service users and roles: `opc` user in Big Data Cloud Service (BDCS), `bdd` user in Big Data Discovery Cloud Service (BDDCS), and Studio admin user. You also need user access to VPN, Oracle Cloud Storage, and Hadoop Cloudera Manager.

Here is a summary of the three layers that comprise user access in BDDCS:

- User access to Oracle Cloud Services.This is the base user access you need, before you can request a BDDCS instance from a sales representative.
- User access to Oracle Cloud Storage, Hadoop and VPN. These are the user accounts required for BDDCS configuration, in addition to the base user account you use to log into BDCS and BDDCS.
- User access to Studio. This is the Studio admin user that you use initially. You configure this Studio admin user when you create a BDDCS instance. You can later add more users within Studio.

User access to Oracle Cloud Services

Before provisioning BDDCS, create a user account in the Oracle Public Cloud (OPC). This is the account for logging into Oracle Public Cloud and into Big Data Cloud Service. At least one Big Data Cloud Service must be provisioned for your account.

You provide this account to your Oracle sales representative, so that they can place an order for BDDCS for you. See Oracle Cloud User Roles and Privileges in *Getting*

Started with Oracle Cloud. You can use this Single Sign-On (SSO) account for logging into Oracle Public Cloud and into the BDCS, once you configure it.

Here is how you use your SSO account with BDDCS:

- For initial access to BDDCS, you cannot rely on SSO. Instead, when an Oracle sales representative places a BDDCS order for you, you receive a "Welcome" email with the URL to provision BDDCS. This way, you are granted initial user access to BDDCS as the service administrator. You then use this access to create your instance.

- After you have created the BDDCS instance, you rely on SSO for logging into all of these accounts: your Oracle Cloud Services Account on Oracle Public Cloud, and into your BDCS and BDDCS instances.

User access to Oracle Cloud Storage, Hadoop and VPN

Before provisioning BDDCS, in addition to the user account in Oracle Public Cloud, you need these user accounts:

- **An Oracle Cloud Storage account**. You specify the storage user name and password when you create a BDDCS instance. You should already have this account. See Managing Storage Credentials in Hadoop.

- **A Hadoop Cloudera Manager administrator account for your BDCS instance**. You fill in this information under **Hadoop Details** when you create a BDDCS instance. This account should have already been created when you created your BDCS instance.

- **VPN access for BDDCS and BDCS instances**. Without this account, you cannot access BDDCS Studio in the BDCS cluster. See

User access to Studio

User access to BDDCS Studio is controlled initially at BDDCS provisioning time. This is when you, as the service administrator, configure user name and password for the Studio admin user. Once BDDCS instance is up and running, you can log into Studio using these credentials. Next, you can change your admin user password in Studio, add more users in Studio, and control their privileges from within Studio.

You cannot integrate with an external LDAP system for managing Studio's users.

If you later change the Studio's database user account, use **Update Credentials** in the BDDCS console to also update this account in BDDCS. However, if you change the password for the Studio admin user (not the Studio's database user account), there is no need to update credentials in the BDDCS console.

For information on adding and managing users in Studio, see the *Big Data Discovery Cloud Service Administrator's Guide*.

Accessing the Console
: For initial access to BDDCS, use the link in the "Welcome" email sent to you for creating a BDDCS instance. For subsequent access, sign into the Oracle Cloud website.

Accessing the service node in a cluster
: To access Big Data Discovery Cloud Service, log into the service node of the Big Data Cloud Service cluster as the `opc` user and change the user to `bdd` to perform any service-specific operations, such as restoring from a backup, or applying patches.

Accessing the Console

For initial access to BDDCS, use the link in the "Welcome" email sent to you for creating a BDDCS instance. For subsequent access, sign into the Oracle Cloud website.

Here is a summary of your options for accessing the service:

- For initial access to BDDCS, when you receive a "Welcome" email for creating a BDDCS instance, you are granted access to BDDCS as the service administrator. You then use this access to create your instance.
- After you have created the BDDCS instance, you rely on Single Sign-On into the Oracle Cloud for accessing all accounts: your Oracle Cloud Services Account on Oracle Public Cloud, Big Data Cloud Service instance, and Big Data Discovery Cloud Service instance.

To access the Big Data Discovery Cloud Service console:

1. Display the Sign In to Oracle Cloud page by clicking the **My Services URL** link in your "Welcome" email or by following these instructions:

   a. Open your web browser and go to the Oracle Cloud website: `http://cloud.oracle.com`

   b. Click **Sign In**.

   c. In the My Services box, select the data center where your services are located.

   d. Click **Sign in to My Services**.

2. On the **Sign In to Oracle Cloud** page, enter your user name, your password and the name of your Identity Domain. Then, click **Sign In**.

   The **My Services Dashboard** is displayed.

3. In the list of services, locate the entry for Oracle Big Data Discovery Cloud Service and then click its name.

   The Oracle Big Data Discovery Cloud Service console is displayed.

Accessing the service node in a cluster

To access Big Data Discovery Cloud Service, log into the service node of the Big Data Cloud Service cluster as the `opc` user and change the user to `bdd` to perform any service-specific operations, such as restoring from a backup, or applying patches.

The Big Data Cloud Service cluster includes the `opc` user account. For information on it, see About Oracle Big Data Cloud Service Users and Roles in *Getting Started with Big Data Cloud Service*.

In addition, Big Data Discovery Cloud Service (BDDCS) runs its operations on one of the nodes in the BDCS cluster as the `bdd` user. This user is created in Big Data Cloud Service when the instance of Big Data Discovery is provisioned for you. This means that if you need to run BDDCS operations directly on its node in the BDCS cluster, you need to change to the `bdd` user.

To log into the BDCS node with BDDCS on it, you should use the private key pair of the public key stored in the `/home/opc/.ssh/authorized_keys` file (on all nodes in the BDCS cluster).

This assumes that the BDCS administrator had already created multiple key pairs for SSH in the BDCS cluster, one of which is for BDDCS use. Coordinate with the BDCS administrator to obtain the public key of the pair you can use on the BDDCS node. For information on how multiple key pairs for SSH are added in BDCS, see Supporting Multiple Key Pairs for Secure Shell (SSH) Access.

To access the BDDCS node in a BDCS cluster:

1. Run the SSH utility to log into the BDCS node as an `opc` user and switch to the `bdd` user. For example:

   ```
   ssh -i ~/.ssh/id_rsa opc@node123bda05 sudo su bdd
   ```

   where: `opc` is the user in BDCS, `node123bda05` is the BDCS node hosting BDDCS in the BDCS cluster, and `id_rsa` is the location of the private key pair for the node.

As a result, you are logged into the BDCS node hosting BDDCS instance as the `bdd` user. This lets you restore BDDCS from a backup, apply an upgrade patch to the BDDCS instance, access management console in Hive, and run the Data Processing (DP) CLI in BDDCS.

Enabling VPN access to the service

To ensure secure access to the service instance by your end users, you must establish an IPSec VPN connection between the Oracle Cloud to the service instance host. The VPN provisioning process is a collaborative effort between Oracle Public Cloud network engineers and your corporate network administrators.

To request VPN, see My Oracle Support Note 2056914.1. Instructions in this topic are complementary to the note in My Oracle Support referenced in the URL.

> **Note:** If you attempt to connect to BDDCS through a connection that is not protected by VPN, Oracle Cloud MyServices prevents such connections and issues an error.

Before you request VPN, ensure these requirements are met at your site:

- **Product-related requirements for VPN**. If you used VPN in Big Data Cloud Service (BDCS), continue to rely on the same VPN routing in BDDCS. If you have not used VPN in BDCS, and now are purchasing BDDCS, you should request VPN from your sales representative.
- **VPN device requirements**. You need a VPN gateway device that uses current IPSec standards to establish a secure tunnel between your network and the Oracle Public Cloud. You will provide the details of your device to Oracle. The device must support:
  - IPv4 traffic with support for ICMP, TCP and UDP. Multicast traffic is not supported.
  - Tunnel mode sessions. Tunnel mode is used to create a virtual private network between your network and the Oracle Public Cloud, rather than between a specific set of hosts. It is used to protect all communications between both networks.
  - Authentication with pre-shared keys. The same pre-shared key is configured on each IPSec VPN gateway device.

  - Dynamic rekeying. IPsec uses dynamic rekeying to control how often a new key is generated during communication. Communication is sent in blocks and each block of data is secured with a different key.

- **Network requirements for an IPSec VPN connection**. Both sides must provide subnets:

  - On your side, dedicate subnets in your network for this VPN connection. You will indicate these subnets to Oracle. To prevent an IP address conflict in the end-to-end network connection, mask your internal systems with a public or non-RFC 1918 address range.

  - On the Oracle side, the network engineers from the Oracle Cloud Operations will provide the destination subnets in a way that avoids IP address conflicts.

To request a VPN provisioning by Oracle Support:

1. Contact your sales representative and ask them to place an order for Oracle VPN Cloud Service. This can be a separate order, or it can be made in conjunction with an order for BDCS and BDDCS (this service).

2. Once you have an active Oracle Cloud VPN Service subscription, go to the My Oracle Support Note 2056914.1 and follow its instructions.

Oracle engineers receive your information and check all prerequisites are met. Next, Oracle provisions the VPN service together with your network engineers during an agreed maintenance window and runs through a post-configuration checklist with you to ensure that the VPN connection is working and that the setup is completed.

Requesting and activating a subscription

To request a subscription to Big Data Discovery Cloud Service, contact your Oracle sales representative and provide the name and email of an account administrator for your service.

Note: Trial subscriptions are not available for Oracle Big Data Discovery Cloud Service, and you cannot purchase a subscription through the web site. You must purchase a subscription from an Oracle sales representative who places an order for you.

Before requesting a subscription ensure that you have:

1. An Oracle.com account (My Account) and the Oracle Cloud Services (MyServices) account.

2. Access to Oracle Big Data Cloud Service (BDCS) and to Oracle Cloud Storage, both of which should already be provisioned in the same Identity Domain.

3. Confirmation that you can connect to these services securely via a VPN connection established by your IT administrator.

For these requirements, see About users and roles and Before you begin with your subscription .

To request a subscription to Oracle Big Data Discovery Cloud Service:

1. Contact your Oracle sales representative and provide an email address of the account administrator for your service.

They place an order for BDDCS in the ordering system in Oracle Public Cloud. If your email was provided, you are granted the account administrator access and receive a "Welcome" email with a link to activate the order for BDDCS.

For simplicity, the next step assumes you serve as the designated account administrator. If you are assigned a service administrator role, then, after you receive a "Welcome" email, you log into Oracle Cloud Services as the service administrator, and start the activation process.

2. Follow the e-mail link to activate a BDDCS instance and provide the service administrator's email.

   It can be the same user who has the account administrator role for BDDCS, or a different user in your organization.

   The Identity Domain administrator in Oracle Public Cloud assigns a BDDCS service administrator role. This sends an email from Oracle Cloud My Services to the designated service administrator (it can be you or another person in your organization).

Now you can log into Oracle Cloud Services to create a BDDCS instance.

Logging into the service console for the first time

After you receive a "Welcome" email about a BDDCS instance enabled for you, log into the Service Console URL to provision your instance.

To log into the BDDCS console for the first time:

1. Click the URL in the "Welcome" email.

   The screen opens to let you log in.

2. Enter the user name of the designated service administrator of BDDCS.

   The credentials for logging in are included in the previous email, when your request for a BDDCS instance was granted. Do not confuse this user with the account administrator for BDDCS. Even though these two users can be the same person, they also can be different.

3. Enter your password for the service administrator of BDDCS.

   Locate this service administrator password for the Identity Domain in the already provisioned Oracle Cloud Storage Instance or Big Data Cloud Service instance. All of them share the same system administrator's user account and Identity Domain.

4. Enter the name of the Identity Domain in which you have an active Oracle Cloud Storage Service and an active BDCS instance.

   This Identity Domain should not have any BDDCS instances already provisioned on it.

5. Click **Sign In**. The BDDCS provisioning wizard opens.

Next, provision your BDDCS instance in three steps.

Creating a service instance

To create a Big Data Discovery Cloud Service instance, use the three steps in the **Create Instance** wizard: select a cluster for hosting the instance, provide configuration details and user credentials for all required underlying services, and confirm your instance configuration.

Step 1: Select a host for your service instance
In the **Service** step of the wizard, select a Big Data Cloud Service cluster to host your instance.

Step 2: Provide service details
In the **Configuration** step of the wizard, provide five different sets of credentials. They include backup details, Hadoop details, WebLogic Administrator, Studio Administrator, and Studio Database User.

Step 3: Confirm your service configuration
In the **Confirm** step of the BDDCS provisioning wizard, confirm your configuration details and all user credentials.

Step 1: Select a host for your service instance

In the **Service** step of the wizard, select a Big Data Cloud Service cluster to host your instance.

Before you can create a Big Data Discovery Cloud Service instance, you must have an active Oracle Big Data Cloud Service (BDCS) account, and an Oracle Cloud Storage account. At least one Big Data Cloud Service must be provisioned for your account. Also, a secure connection through VPN must be available to your Big Data Cloud Service.

Before this step, you should have already received a "Welcome" email. In all steps in the wizard, an asterisk indicates a required field.

To run the first step of the wizard:

1. Click the link provided in the email, log into the Service Console URL, and enter your service administrator credentials provided in the email. The wizard **Configure Big Data Discovery Cloud Service** starts, and the **Service** page is displayed.

2. On the **Service** page, fill in these fields for the BDCS cluster hosting your BDDCS instance:

Field	Description
Big Data Discovery Cloud Service Name	Required. A unique name for your new Big Data Discovery Cloud Service instance. The instance name must be up to 50 characters; it must start with a letter; it can contain only letters, numbers and hyphens (-); it cannot end with a hyphen (-); it cannot contain spaces or underscores. If you enter any invalid values, the field is highlighted in red when you switch to the next step. In this case, enter a new name.
Description	Optional. A description for your BDDCS instance.
Big Data Cloud Service Instance	Required. A Big Data Cloud Service cluster upon which to provision your new Big Data Discovery Cloud Service instance. Select the cluster from the list.

3. Click **Next**. If you entered the values correctly, the second step of the provisioning wizard opens where you configure the details for your service.

Step 2: Provide service details

In the **Configuration** step of the wizard, provide five different sets of credentials. They include backup details, Hadoop details, WebLogic Administrator, Studio Administrator, and Studio Database User.

To provide user credentials and service details:

1. On the **Configuration** page, fill in the fields in each of the five areas.

   As indicated by an asterisk adjacent to a field name, all the fields are required.

2. **Confirm Backup Details**—provide the details for your Storage Cloud Service instance:

Field	Description
Storage Service URL	Specify a valid Storage Cloud Service URL, such as `https://storage.oraclecorp.com/v1/Storage-MyStorage`. To locate it, go to **MyServices** > **Storage Service** > **Service Detail** > **Service REST Endpoint**.
Storage Container	Specify the location for storing backup files in your Storage Cloud Service instance, such as `bddcs_backup`. If the storage container you specify does not exist, it is automatically created.
Storage User Name	Enter the user name for accessing your Oracle Cloud Storage Service account. Use the format `username[.role]`. For example, enter `myuserid.StorageAdmin`.
Storage Password	Enter the password associated with the Cloud Storage Service account user name. For information, see Managing Storage Credentials in Hadoop.

3. **Confirm Hadoop Details**—provide the details for the Hadoop configuration of your BDDCS instance:

Field	Description
User name	Enter a new user name for the Cloudera Manager in Hadoop.
Password	Enter a password for the Cloudera Manager in Hadoop.
Hive Database Name	Select the name of the Hive database you want Big Data Discover Cloud Service to regularly scan for data.

4. **Create WebLogic Administrator**—fill in the credentials for accessing the WebLogic Server Administrator (this is the container for BDDCS Studio):

Field	Description
User name	Enter a new WebLogic Server Administrator user name for BDDCS.
Password	Enter a password for the WebLogic Server Administrator for BDDCS. It must contain at least 8 characters, one of which must be a number, and cannot start with a number.

Field	Description
Confirm Password	Confirm the password.

5. **Create Studio Administrator—**fill in these fields for Studio's credentials:

Field	Description
Email address	Enter the valid email address of the Studio Administrator, which will be their user name. This must be a full email address. It cannot begin with `root@` or `postmaster@`.
Password	Enter the password for your Big Data Discovery Cloud Service Studio Administrator. The password must be between 8 and 30 characters, with at least one lower case letter, one upper case letter, one number and one special character (_, -, !, @, #, $, ^, (), [], {}, :, +, ~), and cannot be an ordinary word. For example: **Welc9me!23**
Confirm Password	Confirm the password.

6. **Confirm Studio Database User—**fill in these fields for the Studio database user in BDDCS:

Field	Description
User name	Enter a new Studio database user name.
Password	Enter a password for your Studio database user.
Confirm Password	Confirm the password.

7. Click **Next**.

   If you entered the values correctly, the last step of the provisioning wizard opens. In it, you confirm the information you have provided.

Example 1-1 Example

Here is an example of credentials for the **Configuration** screen of the BDDCS provisioning flow. They are grouped in five sections:

```
#Backup details:
Storage Service URL: https://<url_provided_by_Oracle_CLOUD>.com/my-storage
Storage Container: bddcs_my_backup
Storage User Name: my.Storageadmin
Storage Password: my-password1

# Hadoop Details:
User Name: my-admin
Password: admin
Hive Database Name: default

# WebLogic Administrator details:
Create WebLogic Administrator
User Name: weblogic
Password: password123
Confirm Password: password123

# Studio Administrator details:
Email Address: user1.acme.com
```

```
Password: Welc9me!23
Confirm Password: Welc9me!23

# Studio Database User
User Name: studio
Password: Welcome!23
Confirm Password: Welcome!23
```

Step 3: Confirm your service configuration

In the **Confirm** step of the BDDCS provisioning wizard, confirm your configuration details and all user credentials.

To confirm your Big Data Discovery Cloud Service settings:

1. In the wizard, click **Confirm**. The resulting screen shows you the details of the service you are about to provision.
2. If satisfied, click **Create**.

   Your BDDCS instance provisioning starts in the Oracle Cloud. The dashboard for BDDCS opens. It includes the status of your service: `Enabling`.
3. Check whether **Service Status** changes to `Up`.

   It takes about one hour to finish the service provisioning. The system refreshes its status automatically every 5 seconds when it runs the enabling operation. You can also click the ⟳ icon to see the current status.

Once the status changes to `Up`, the BDDCS instance is successfully provisioned for you.

> **Note:** If there is a problem and the status changes to `Enable Failed`, click **Re-enable** in the life cycle operations ≡ icon. If the status changes to `Unknown`, click the refresh ⟳ icon. If the problem is transient, the BDDCS instance recovers. If the problem persists, contact Oracle Cloud Support.

Next, to continue using the service securely, replace the SSL certificates in WebLogic Server, to use your own set of SSL certificates. You can also review the status of your instance, configure backups, and access Studio.

Replacing SSL certificates

Your BDDCS instance is provisioned in the Oracle Cloud with SSL enabled in WebLogic Server by default. Enabling SSL for BDDCS activates WebLogic Server's default Demo Identity and Demo Trust Keystores. As their names suggest, these keystores are untrusted and meant for demo purposes only. After creating a BDDCS instance and running the provisioning wizard, you should replace them with your own certificates.

For information on WebLogic's demo keystores, see the section Configure keystores in the *WebLogic's Administration Console Online Help*.

Verifying the service instance

To verify all BDDCS components are running successfully, log into Studio and add a data set by uploading a CSV file. Observe it in the Studio's Catalog and **Explore** pages.

It is useful to distinguish between these two user interfaces in BDDCS:

- In the BDDCS console, you can start and stop the instance and perform other life cycle operations, such as configure and run backups. You also log into Studio from here.
- In Studio, you load, explore, transform, and visualize data. (To automate loading data, use the Data Processing CLI in BDDCS).

You can create a new data set in Studio by uploading personal data from files. Studio supports Microsoft Excel files, delimited files such as CSV, TSV, and TXT, and also compressed file such as ZIP, GZ and GZIP. A compressed file may include only one delimited file. After upload, the data is available as a data set in the Studio's Catalog.

To verify your BDDCS instance:

1. In the BDDCS console, confirm that the status of the instance is `UP`. This indicates that all BDDCS components are running.
2. Find any publicly available CSV file of relatively small size. A file of the size of several MB works well.
3. Copy the file to local storage on any BDCS node, and then use the HDFS `put` command to copy it into HDFS.
4. Log into Studio. Click **Open BDDCS Studio** and provide your Studio admin user's email and password. See Accessing Studio.
5. In Studio, load a CSV file:
   a. Go to Studio's Catalog and click **Add Data Set**.
   b. Click **Create a data set from a file**, and then click **Browse**.
   c. Locate the file, and click **Open**, and then click **Next**.
   d. In the **Preview** page, make changes as needed. For example, you can exclude an attribute from the data set, modify the name of the attribute, specify the header row, optionally omit rows from an uploaded file (this will limit the data), and also specify delimiters, quote signs, language, and encoding settings.
   e. Click **Next**. After **Create you data set** opens, specify its details and click **Create**. Studio creates a new data set based on the uploaded file. Studio maps the data set name to a unique Hive table name that BDD creates for it. The data set appears in the Catalog.

As a result, you have added a data set to BDDCS Studio. If the file uploads successfully, this confirms that the Data Processing component of Big Data Discovery is running successfully. The Dgraph and the HDFS Agent components are also running successfully.

Now that you have loaded your first data into BDDCS using Studio, learn how to use Studio to explore and analyze your data, or how to load more data and update it.

Accessing documentation for the service

Find BDDCS documentation in the Oracle Help Center, under the Cloud tile. Use BDDCS Getting Started Guide (this document) to learn how to provision and use BDDCS.

To access BDDCS documentation:

1. Go to **Oracle Help Center**: http://docs.oracle.com/en/
2. Select the **Cloud** tile: http://docs.oracle.com/cloud/latest/
3. Select the **Platform and Infrastructure** tab, and then select **Big Data Discovery Cloud Service**.

The main page for Big Data Discovery Cloud Service documentation opens. From here:

- To provision, start, stop, back up, manage BDDCS, and learn how to use BDDCS and Studio, see the *BDDCS Getting Started Guide* (this document).
- For logging and troubleshooting BDDCS, Dgraph, and Studio, see the *BDDCS Administrator's Guide*. For logging on Data Processing component, see the *BDDCS Data Processing Guide*.
- To learn how to use Studio to explore, transform, and discover data, see the *BDDCS Studio User's Guide*.
- For information on Data Processing component in BDD, interaction with Hadoop, and Data Processing CLI for automated loading and updating of data in BDDCS, see the *BDDCS Data Processing Guide*.
- To create custom visualizations in Studio, see the *BDDCS Extensions Guide*.
- For information on EQL, the query language for creation of custom views in Studio, see the *BDDCS EQL Reference*.

2

Instance Life Cycle Tasks

This section includes all life cycle operations for the BDDCS instance. You can see the status, start, stop, run a backup on demand, configure scheduled backups, and restore from backups. You can also apply a patch, reset and re-enable your instance if provisioning fails, and update credentials for the Studio database and WebLogic Server.

The Overview page
: Use the **Overview** page to view the current status of your BDDCS instance, obtain the host name of the BDCS node that is hosting your BDDCS instance, and check subscription ID, service start and stop dates.

The Administration page
: The **Administration** page has two tabs: **Backup** and **Maintenance**. It lets you configure, run and view backups and access the page for downloading BDDCS patches.

Starting and stopping an instance
: After you provision a BDDCS instance, it is automatically started. In the dashboard, you can stop and restart an already provisioned instance.

Accessing Studio
: You access Studio by using **Open BDDCS Studio** button in the BDDCS console. Studio is the front-end application for your BDDCS instance.

Scheduling backups
: To configure a backup, specify the automated backup frequency as backup policy. Also specify how long you want to retain the backup data and the location of your backup storage.

Creating a backup on demand
: Use **Back up Now** to run a backup on demand.

Restoring an instance from a backup
: You can restore your BDDCS instance from backup using the `bdd-cloud.sh` script.

About loading data into your service
: You can load data either in Studio by uploading a personal file, or you can run DP CLI to load and update data in the service.

Patching the service
: When a patch is available, the **Maintenance** tab displays a notification and a link to download the patch files. Once you have the files, you can copy them to your node and apply the patch. You can also apply WebLogic patches at the same time.

Rolling back a failed patch

If the patch fails, you should run the rollback script to remove the patch and restore your old instance. You can then fix the errors that caused the failure and reapply the patch.

Resetting and re-enabling an instance

If enabling of the BDDCS instance fails, the life cycle operations menu displays two options to help you recover: **Reset** and **Re-enable**.

Updating credentials

In some instances, for example if you changed credentials outside of the BDDCS instance (for Hadoop, Studio, or WebLogic Server), you need to update these credentials in the BDDCS instance configuration to reflect the changes.

Deleting an instance

When you delete an instance, an automatic backup of your configuration is created and your configuration is removed.

Termination

When your subscription period is close to expire, you, as a service administrator, receive a notification email from Oracle Cloud Services and the service is terminated at the expected date. To continue with your subscription and prevent the termination, contact your Oracle Service representative before your subscription expires.

The Overview page

Use the **Overview** page to view the current status of your BDDCS instance, obtain the host name of the BDCS node that is hosting your BDDCS instance, and check subscription ID, service start and stop dates.

Overview always shows up by default, when you successfully provision your BDDCS instance.

The page indicates the date and time when it was last refreshed. Click ⟳ to refresh the page. If you run any life cycle operations, such as start, stop, enable an instance, or take a backup, the page refreshes automatically every 5 seconds to indicate the most recent status of the operation that is being run.

Element	Description and contents
Oracle CLOUD MyServices	Shows the service that is currently open. Includes: • The currently logged in user • The domain in which the service is running • The expandable arrow for Help, Accessibility, the **About** information, and a link to **Sign Out**.

Big Data Discovery Cloud Service	The heading for this service. If you expand it, it may show, for example:
	`Service Name:my-instance-1905`
	`Description:my-cloud-instance`
	`Subscription:PRODUCTION`
	`Expires:25 Apr 2017 03:36 PM GMT`

Element	Description and contents
☰	The icon for accessing life cycle operations for the service: • Open BDDCS Studio • Stop the service and restart it • Delete the instance • Update credentials for WebLogic Server Administrator, Studio and Hadoop. • Reset and re-enable the service (displayed only if provisioning fails).
Open BDDCS Studio	The **Open BDDCS Studio** link. Studio is the front-end application for Big Data Discovery Cloud Service, where you see your source data, explore it, transform it, and create visualizations.
Overview	The link to **Overview** (this page). It is already opened.
Administration	The link to the **Administration** page for running and configuring backups and applying a patch.
Service Status	The **Service Status** can be one of the following: `Up`, `Down`, `Enabling`, `Backing up`, `Disabling`, `Stopping`, `Stopped`, or `Enable Failed`, `Unknown`. If the service is running, the time since the service is up is also listed. if you are recovering from an error and Oracle Cloud Support has been helping you, then even if the system may be back up again, you may still see the error displayed on your screen. The system status supersedes the error message you may still observe. To recover, stop and restart the instance (do not use **Reset**). If the status is `Unknown`, the **Overview** page does not show provisioned users, versions of BDCS and BDDCS, and the time since the service was up. Action items in the life cycle operations menu are also not available. The `Unknown` status displays when a communication problem occurs due to a network issue, or a missing response from the BDCS cluster nodes. To get out of the `Unknown` state, click the refresh ↻ icon. If the problem is transient, the BDDCS instance recovers. If the problem persists, contact Oracle Cloud Support.
Provisioned Users	The number or licensed users for this BDDCS instance.
Licensed Users	The number of users currently provisioned for this instance.
BDD Nodes	The number of BDD nodes. In this release, BDDCS runs on a single node. This is the BDDCS node in the Big Data Cloud Service (BDCS) cluster.
Total BDA Nodes	The total number of licensed BDCS nodes. These are Hadoop nodes in the BDCS cluster.
↻	The icon for the date and time of the last refresh operation for the service. The page refreshes every 5 seconds, or you can refresh it by clicking this icon.

Element	Description and contents

BDD Nodes	The details of **BDD nodes**.
	Includes the hostname of the BDCS node hosting the BDDCS instance. You use this hostname to access the hosting node in BDCS cluster, restore, and apply a patch to your BDDCS instance. For example:
	`bdd.BDCShostname`
	`Version: 1.2.x.x.x`
	If the number of nodes changes (due to adding more Hadoop nodes to the Big Data Cloud Service cluster), the overview page reflects that. For information on adding more nodes in the Big Data Cloud Service cluster, see Adding Nodes to a Cluster.

Big Data Cloud Service Instance	The details of the Big Data Cloud Service Instance, such as the cluster name, and the version of the BDCS software.For example:
	`serverABCD`
	`Version: 4.4.x`
	`Nodes: 6`

Additional Information	**Additional information**. Includes the name of your Oracle Cloud plan, the start and end dates of your service, the subscription ID and the order ID. For example:
	`Plan: Big Data Discovery Cloud Service`
	`Service Start: 21 December 2015 03:36 PM GMT`
	`Service End:   21 December 2016 03:36 PM GMT`
	`Subscription ID: 123456789`
	`Order ID: 5071999`

The Administration page

The **Administration** page has two tabs: **Backup** and **Maintenance**. It lets you configure, run and view backups and access the page for downloading BDDCS patches.

The Backup tab

The **Backup** tab opens by default. It shows the list of backups, and lets you configure backups or run a manual backup on demand. If the list is long, you can scroll it by selecting pages, or back and forward arrows. The list of already created backups is also displayed, it includes both scheduled backups and all manual backups that you run. For example, for a specific backup, you may see this information:

```
03/23/2016 09:39PM GMT
Type: Manual
Available Until: 03/23/2016
File name: BDD_BACKUP_20151123_2140001.tar
File size: 1.3 Mb
```

To delete a specific backup, click the ☰ icon to the right of the backup's listing.

The Maintenance Tab

The **Maintenance** tab informs you where to download BDDCS patches. It states:

```
Download and apply the latest BDDCS patch set release from http://aru.us.oracle.com/
```

See Patching the service.

Starting and stopping an instance

After you provision a BDDCS instance, it is automatically started. In the dashboard, you can stop and restart an already provisioned instance.

You can run only one life cycle operation at a time. When an operation is running, you cannot access BDDCS Studio.

To stop and start a BDDCS instance:

1. In the life cycle operations menu ☰, select the option you need:

   a. To stop, select **Stop Instance**. The system asks if you are sure you want to stop this instance. If you click **OK**, the request for stopping the instance is accepted. You cannot run any other operations until this operation completes. The service status changes to `Stopping` and then to `Stopped`.

   b. To start, select **Start Instance**. If the instance exists, the system starts it.

Accessing Studio

You access Studio by using **Open BDDCS Studio** button in the BDDCS console. Studio is the front-end application for your BDDCS instance.

Before you can access Studio, you need to create the BDDCS instance and establish a VPN connection to it. You should also have configured access to the Studio interface, as the Studio admin user.

To open BDDCS Studio:

1. In the BDDCS console, click **Open BDDCS Studio** in the upper right corner. The Studio's **Welcome** page displays.

2. Type your email address or user name in the top field, as prompted. A Studio deployment is configured to take one value or the other. Check the field label to verify that you entered the valid information.

3. In the **Password** field, type your password for the BDDCS Studio Admin user. To save your login information in the current browser, check the **Remember me** check box.

4. Click **Sign in**.

The Studio application opens. To learn about Studio, see Taking a Tour of Studio.

Scheduling backups

To configure a backup, specify the automated backup frequency as backup policy. Also specify how long you want to retain the backup data and the location of your backup storage.

You can either schedule backups to run automatically, or back up you instance at any time on demand.

You can run a backup on both a running BDDCS instance and on the instance that is stopped.

When the backup operation runs, it does not change the state of the instance. That is, if backup starts when the instance is running, it won't stop the instance, and if backup starts when the instance is down, the instance remains down when backup completes.

While a backup runs, you cannot access BDDCS Studio, or perform any other operations on your instance.

If you don't change any settings in the **Backup Configuration**, the system uses a default backup policy. It uses these settings:

- The default backup of BDDCS is created automatically each Monday, at 2 a.m. UTC.
- The default backup retention period is 30 days.
- The default backup location is the Cloud Storage Container you specified when provisioning your BDDCS instance.

To schedule backups:

1. In the BDDCS console, go to **Administration** > **Backups** > **Backup Configuration**.

   The **Backup Configuration** screen opens.

2. Enter or select information in the fields as follows:

Field	Description
Backup Policy	Select the **Daily**, or **Weekly** policy, and then select the day of the week and time of the day. This configures the frequency of your service backups. If you don't change this setting, the default backup is created each Monday at 2 a.m. UTC time.
Time Zone	Select country and then time zone. After you select the time zone, the time abbreviation in the field above automatically changes to reflect it.
Data Retention	Select the number of days, weeks, or months to retain your BDDCS backup files. If you don't specify a custom retention period (or don't configure a backup policy), the default retention period of 30 days is used.

   The name of the **Storage Instance**, the **Storage Container** name, and the user credentials for the Oracle Cloud Storage user are already filled in from your previous configuration.

3. Click **Save**. The **Administration** page displays. It indicates that automated backup is configured and lists the details. When the automated backup runs, it is added to the list of backups on the page.

Now you can let Oracle Cloud run backups for you. Alternatively, you can also run a manual backup at any time, by using **Back up Now**.

Creating a backup on demand

Use **Back up Now** to run a backup on demand.

This saves your backup in the Cloud Storage Container you specified when provisioning your BDDCS instance. The backup files are saved for the duration of the retention period if it is configured, or for the default retention period of 30 days.

You can run a backup on both a running BDDCS instance and on the instance that is stopped. When the backup operation runs, it does not change the state of the instance. That is, if backup starts when the instance is running, it won't stop the instance, and if backup starts when the instance is down, the instance remains down when backup completes.

While a backup runs, you cannot access BDDCS Studio, or perform any other operations on your instance.

To create a backup on demand:

1. In the BDDCS console, go to **Administration** > **Backups** > **Back up Now**.

   The **Back up Now** screen opens. The Storage Container name is already specified.

2. In **Keep Forever**, select:

Option	Description
Yes	To keep this backup forever (for the duration of your BDDCS subscription).
No	To keep this backup only for the duration of the backup policy (if it is specified). When the time specified in the backup retention expires, this backup is deleted. If the backup policy does not exist, the backup is stored for the duration of 30 days. This is the default retention period.

3. Optionally, add **Notes** to keep a record of the backup details.

4. Click **Back up Now**. The **Administration** page displays the status `Backing Up...`

   When the backup completes, the **Administration** page displays the time stamp for this most recent backup. The backup is also listed at the top of the backups list.

Restoring an instance from a backup

You can restore your BDDCS instance from backup using the `bdd-cloud.sh` script.

> **Important:** This requires you to SSH into the BDDCS node as the `opc` user, then switch to the `bdd` user. For information, see Accessing the service node in a cluster.

Before restoring, verify that you have the following:

- The hostname of your BDDCS node. This is listed on the **Overview** page under **BDD Nodes**.
- A BDDCS backup .tar file. You should know the name of the file you want to restore from beforehand. This is listed on the **Backup** tab of the **Administration** page.

To restore your BDDCS instance from backup:

1. Open a command prompt and SSH into the BDDCS node as the `opc` user:

   ```
   ssh -i ~/.ssh/id_rsa opc@<hostname>
   ```

   Where *<hostname>* is the full hostname of the BDDCS node.

2. Switch to the `bdd` user:

   ```
   sudo su bdd
   ```

3. Verify that the following environment variables are set:

Environment variable	Value
BDD_WLS_USERNAME	The username for the WebLogic admin
BDD_WLS_PASSWORD	The password for the WebLogic admin

   To determine whether they're set, run:

   ```
   echo $<ENVIRONMENT_VARIABLE_NAME>
   ```

   The above command should print the value of the specified environment variable to the console. If it doesn't, set the environment variable:

   ```
   export <ENVIRONMENT_VARIABLE_NAME>=<value>
   ```

4. Go to `/home/bdd/Oracle/Middleware/BDD` and run the restore script:

   ```
   cd /home/bdd/Oracle/Middleware/BDD
   ./bdd-cloud.sh backup restore --archive_name=<backup_file>
   ```

   Where *<backup_file>* is the name of the file you want to restore from, including the .tar file extension.

The script stops your BDDCS instance, restores it, then restarts it. This may take a long time, depending on the amount of data you have. When the script completes, your BDDCS instance will be fully restored and running.

About loading data into your service

You can load data either in Studio by uploading a personal file, or you can run DP CLI to load and update data in the service.

For an overview of data loading and data update options, see Data Loading and Updates.

To configure DP CLI and use it for loading data, see the *Big Data Discovery Cloud Service Data Processing Guide*. Before you run DP CLI, you must set up a few options in its configuration.

To load data in Studio, see the *Big Data Discovery Cloud Service Studio User's Guide*.

Patching the service

When a patch is available, the **Maintenance** tab displays a notification and a link to download the patch files. Once you have the files, you can copy them to your node and apply the patch. You can also apply WebLogic patches at the same time.

Important: To apply the patch, you must SSH into the BDDCS node as the `opc` user, then switch to the `bdd` user. For information, see Accessing the service node in a cluster.

You apply the patch by running a single script on your BDDCS node. In addition to patching your instance, the script makes a backup of your current instance, which you can use to roll back the patch if it fails. When the script completes, your instance will be completely updated and running.

To patch your BDDCS instance:

1. Go to the **Maintenance** tab of the **Administration** page and click the link for the BDDCS patch.

   The patch request on the Oracle Automated Release Updates site displays.

   Note: On this page, there is also a heading **Available Patches**. This is a placeholder for patches that will be available for the future release. A heading **Patch History** is present but is also empty.

   ---

2. Download all of the patch files in the table at the bottom of the page.

   The files are named `p<`*`patch_number`*`>_Generic_XofX.zip.`

3. If you want to patch WebLogic Server:

   a. Log in to My Oracle Support and click the **Patches & Updates** tab.

   b. Use the **Patch Search** section to locate and download the patch you want.

4. Open a command prompt and SSH into the BDDCS node as the `opc` user:

   ```
   ssh -i ~/.ssh/id_rsa opc@<hostname>
   ```

   Where *`<hostname>`* is the hostname of the BDDCS node. This is listed on the **Overview** page under **BDD nodes**.

5. Switch to the `bdd` user:

   ```
   sudo su bdd
   ```

6. Go to `/opt/oracle/bdd` and create a new directory called `<patch_number>`:

   ```
   cd /opt/oracle/bdd
   mkdir <patch_number>
   ```

   This should be the number from the patch filenames.

7. Copy the patch files from your local machine to `<patch_number>`:

   ```
   scp -r <user>@<local_hostname>:</path/to/BDDCS/patches> /opt/oracle/bdd/
   <patch_number>
               scp -r <user>@<local_hostname>:</path/to/WLS/patches> /opt/oracle/bdd/
   <patch_number>/WLSPatches
   ```

   Where:

   - *`<user>`* is your Linux username

- *<local_hostname>* is your local machine's hostname
- *</path/to/BDDCS/patches>* is the absolute path to the location of the BDDCS patches on your machine
- *</path/to/WLS/patches>* is the absolute path to the location of the WebLogic Server patches on your machine

8. Go into the <patch_number> directory and extract the patch tools:

```
cd <patch_number>
unzip bddcs_tools.zip
```

This creates a directory called /bddcs/bin, which contains the scripts that install the patch.

9. Set the following environment variables:

Environment variable	Value
BDD_HADOOP_UI_USERNAME	Username for the Cloudera Manager admin
BDD_HADOOP_UI_PASSWORD	Password for the Cloudera Manager admin
BDD_WLS_USERNAME	Username for the WebLogic Server admin
BDD_WLS_PASSWORD	Password for the WebLogic Server admin
BDD_STUDIO_ADMIN_USERNAME	Username for the Studio admin
BDD_STUDIO_ADMIN_PASSWORD	Password for the Studio admin

These are required by the upgrade script. You can set them by running:

```
export VARIABLE_NAME=<value>
```

10. Go to /opt/oracle/bdd/<patch_number>/bddcs/bin and run the patch script:

```
./bdd-cloud.sh upgrade bdd --old_bdd_dir=/home/bdd/Oracle/Middleware/BDD --
new_bdd_package_dir=/opt/oracle/bdd/<patch_number>
```

When the patch script completes, your instance is fully updated and running. The **Overview** page should display the new version number. Note that your system still contains some files and directories associated with the previous version of BDDCS; you can remove these if you want.

If the patch fails, you should roll it back and restore your previous cluster. For more information, see Rolling back a failed patch.

Rolling back a failed patch

If the patch fails, you should run the rollback script to remove the patch and restore your old instance. You can then fix the errors that caused the failure and reapply the patch.

The rollback script restores your old instance from the backup file created by the patch script. Before running the script, you should know the absolute path to this file, which is defined by the BDD_BACKUP_FILE property in the bdd-cloud-cron-env.properties file.

To roll back a failed patch:

1. If you haven't already, log into the BDDCS node as the `opc` user and switch to the `bdd` user.

   Instructions for doing this can be found in Patching the service.

2. Go to the location of the rollback script:

   ```
   cd /opt/oracle/bdd/<patch_number>/installer
   ```

3. Run the rollback script:

   ```
   ./rollback.sh bdd.conf
   ```

4. Enter the absolute path to the backup file you want to restore from (including the filename and extension) when prompted.

   This is defined by the `BDD_BACKUP_FILE` property in the `bdd-cloud-cron-env.properties` file.

When the script completes, your old instance will be up and running. You can then fix the errors that caused the patch script to fail and rerun it.

Resetting and re-enabling an instance

If enabling of the BDDCS instance fails, the life cycle operations menu displays two options to help you recover: **Reset** and **Re-enable**.

If enabling of the BDDCS fails, the Overview page displays an `Enable failed` state. In this case, you have two options:

- Click **Re-enable** to retry enabling the instance.
- Click **Reset** to return back to the first step of the provisioning wizard.

If the BDDS enablement succeeds, these options do not appear in the life cycle operations menu. Also, depending on the nature of the failure, the **Reset** option might not be available in the menu, and only **Re-enable** is available.

To reset or re-enable the BDDCS instance:

1. Right-click the cycle operations icon ☰, and select the option you need:

   a. To re-enable, select **Re-enable**. This tries to enable the instance again.

   b. To reset, select **Reset**. This returns you back to the first step in the BDDCS provisioning wizard.

      See Creating a service instance, to repeat the provisioning process.

Updating credentials

In some instances, for example if you changed credentials outside of the BDDCS instance (for Hadoop, Studio, or WebLogic Server), you need to update these credentials in the BDDCS instance configuration to reflect the changes.

To update credentials:

1. Go to the BDDCS dashboard in Cloud MyServices, and locate the life cycle operations icon: ☰

The list shows operations you can perform, such as open Studio, start, stop and delete the instance, and update credentials.

2. Select **Update Credentials**.

   The window for updating credentials displays.

3. Provide these WebLogic Server Administrator details:

Field	Description
User Name	Enter the Weblogic Server user name for the Big Data Discovery Cloud Service.
Password	Enter the password for your WebLogic Server Administrator user in the BDDCS instance.

4. Provide Studio Database Administrator details:

Field	Description
User Name	Enter the Studio database user name for the Big Data Discovery Cloud Service.
Password	Enter the password for your Big Data Discovery Cloud Service Studio database user.

5. Provide these Hadoop details:

Field	Description
Cloudera User Name	Enter the Hadoop user name for the Big Data Discovery Cloud Service.
Password	Enter the password for your Hadoop user.

6. Click **Save**.

Deleting an instance

When you delete an instance, an automatic backup of your configuration is created and your configuration is removed.

If your subscription is valid, you can create a new BDDCS instance at any time after you delete an old one. To create a new instance, re-enable it, and then restore your configuration from backup.

Here are the actions that take place when you delete the BDDCS instance:

- An automatic backup of your BDDCS index files is created. If you later re-enable and start a new BDDCS instance within this BDDCS account, you can run the restore script with this backup.
- All internal files related to BDDCS are removed from the hosting BDCS node. The data bases (indexes) produced by the Dgraph and the Studio database files are removed. This means that Studio projects are deleted from Studio. The Hive and AVRO files are not removed on the Oracle Cloud Storage. Remember that these might be the files that BDDCS creates, when it creates data sets.
- If you provision a new instance after deleting this instance, before you run DP CLI, examine your Hive tables and clean any that you don't want to be re-provisioned when you run the DP CLI. For example, if you see any files created by BDDCS automatically from previous runs of a BDDCS instance, remove them.

- Files uploaded into BDDCS via a personal data upload in Studio are saved by BDDCS in HDFS running on the BDCS cluster instance. These files are not removed. You should remove them. If you later start a new BDDCS instance, you can load these files into Studio again if needed, using the Data Processing CLI.

You do not need to stop the instance before deleting it.

To delete a BDDCS instance:

1. Go to the BDDCS console in Cloud Services, and locate the life cycle operations icon: ☰

2. Select **Delete Instance**.

   The warning message displays:

   ```
   You are about to delete the BDDCS instance.
   This will remove all BDDCS software components
   from the BDD node but will not remove any data
   from the BDCS cluster. Are you sure you want to
   delete the BDDCS instance <instance-name>?
   ```

3. Click **OK** to delete an instance.

   When the instance is deleted, Oracle Cloud Services runs a backup procedure, and starts disabling the instance. The **Overview** page first shows the status `Disabling`, and then returns you to the first step of the provisioning wizard where you can re-provision a new BDDCS instance, using the same BDDCS account if your subscription is valid.

Termination

When your subscription period is close to expire, you, as a service administrator, receive a notification email from Oracle Cloud Services and the service is terminated at the expected date. To continue with your subscription and prevent the termination, contact your Oracle Service representative before your subscription expires.

The support team for the Oracle Public Cloud Service is responsible for terminating a BDDCS instance. You cannot terminate the instance yourself.

Termination of the instance results in these actions by the Oracle Public Cloud Service:

1. Creation of an automatic backup of existing data and metadata (including the BDDCS index) and saving it as a backup in the Cloud Storage Service.

2. Deletion of the BDDCS instance in the BDCS cluster instance.

> **Note:** The BDDCS instance is deleted as soon as the subscription expires. If you want to renew the service, do it before your subscription expires.

3

Welcome to Big Data Discovery

This section introduces Big Data Discovery (BDD), and includes overviews of its components.

Ways of using Big Data Discovery

In your organization, use BDD as the center of your data lab, as a unified environment for navigating and exploring all of your data sources in Hadoop, and to create projects and BDD applications.

Using BDD in the data lab

The data lab is a center of innovation that makes analytics creativity possible for everyone who works with data. It supports a large portfolio of data projects, and is integrated with the production environment for commercialization and feedback.

The data lab is a complete set of descriptive, diagnostic, predictive, and prescriptive analytics solutions. It is a place for collaboration between a select group of analysts who work together as a team and have easy access to multiple data sets. The data lab serves as an easy sandbox for ad-hoc data experiments.

When used in the data lab, Big Data Discovery lets you:

- Define projects in BDD and share them with others in your research data lab.
- Use BDD to quickly develop ideas, build prototypes and models, and invent ways of deriving value from data. For example, you can try out new approaches, quickly discard the ones that are not working, and move on to try new ways of working with your data.
- Shape data in your projects in multiple ways, from making easy single-row changes, such as trimming, editing, splitting, or null-filling, to advanced data shaping techniques, such as aggregations, joins, and custom transformations.
- Let others in your organization consume models created in the data lab, and publish insights to decision-making groups inside your organization.

Using BDD to navigate and explore data sets in Hadoop

When used as the navigator on top of your data sources in Hadoop, BDD visually represents all data available to you in your environment. You see all the data in Studio's Catalog and can find interesting data sets quickly. You can filter, edit metadata on data sets, and create new data sets for others in your team.

Using BDD to create BDD applications

BDD lets you create **BDD applications** for well-established and known problems, to suit your business needs. For example, you can:

- Create BDD projects from scratch, improve them, and share them with wider groups of users in the organization.
- Serve as the author of all discovery solution elements: data sets, information models, discovery applications, and transformation scripts.
- Run discovery repeatedly and transparently to others in your group, and on different large-scale data sets arriving periodically from various sources.

Value of using Big Data Discovery

In BDD, a wider number of people can work with big data, compared with traditional analytics tools. You spend less time on data loading and updates, and can focus on actual data analysis of big data.

In traditional analytics projects you spend most of your effort on data preparation. You must first predict all possible business questions, model the data to match, locate and get the data sources, and manipulate feeds to fit into the model. You build pipelines for extracting, transforming and loading data (ETL). Only after these tasks you can engage with the data. As a result, only a minimal effort is actually focused on data analysis.

The complexity of big data compounds the up-front costs of data loading. Big data increases the amount and kinds of data. You need more time to understand and manipulate new source data, especially unstructured sources, before it is ready for analysis.

Big data also changes fast. You must update your existing analytics projects with newer data on a regular basis. Such data loading and update tasks need qualified engineers with expert skills. As a result, you spend more time and money on up-front data loading and data updates.

Big Data Discovery addresses these problems:

- More users can engage with the data and contribute to big data analytics projects.
- Data preparation becomes a small part of your effort. You don't have to prepare data up-front, and can focus your time and effort on analyzing data, and improving your business.
- You can update BDD projects with new data, refresh already loaded data, or add newer data to existing projects.
- You can work with data as the data scientist, and rely on a range of ways for data shaping and visual analysis in BDD.

Addressing your goals and needs

Big Data Discovery makes your job faster and easier than traditional analytics tools. This topic discusses your goals and needs in data analysis and shows how Big Data Discovery can address them.

As a data scientist or analyst, you:

- **Solve complex questions using a set of disjointed tools**. You start with many dynamic imprecise questions. You often have unpredictable needs for data, visualization, and discovery capabilities. To address them, you rely on open source and custom discovery tools. You use tools in combination with other tools, and often need to reopen the same tools several times.

  In Big Data Discovery, this fragmented workflow between tools is replaced with a single workflow that is natively part of your ecosystem.

- **Need to collaborate**. Together with your team, you work with big data from many external and internal sources. Other team members may consume the results of your findings. You also improve and publish insights and prototypes of new offerings.

  In Big Data Discovery, you can create personal projects, or create and share a project with your team.

- **Want to make sense of data**. You often need to create and use insights. You do this by collecting, cleaning and analyzing your data.

  Big Data Discovery lets you make sense of data. You use it to collect, join, shape, and analyze your data.

- **Generate ideas and insights**. You want to create insights that lead to changes in business, or you want to enhance existing products, services, and operations. You also need to define and create prototypes (or models) for new data-driven products and services.

  In Big Data Discovery, you arrive at insights by using many data visualization techniques. They include charts, statistical plots, maps, pivot tables, summarization bars, tag clouds, timelines, and others. You can save and share results of your discovery using snapshots and bookmarks.

- **Validate, trace back, tweak, and share your hypotheses**. You often need to provide new perspectives to address old problems. The path to the solution is usually exploratory and involves:

  – Hypotheses revision. You must explore several hypotheses in parallel. They are often based on many data sets and interim data products.

  – Hypotheses validation. You need to frame and test hypotheses. This requires validation and learning of experimental methods and results done by others.

  – Data recollection and transparency. You would like all stages of the analytical effort to be transparent so that your team can repeat them. You want to be able to recreate analytical workflows created before. You also would like to share your work. This requires linear history of all steps and activities.

  Big Data Discovery helps you by saving your BDD projects, data sets, and transformation scripts. This lets you improve your projects, share them, and apply saved transformation scripts to other data sets.

Results of working with Big Data Discovery

In BDD you can create an insight, an exploratory project, or a tool for long-term analysis of your data.

For example, you can create:

- Another data set that illustrates the business hypothesis, or answers a specific business question.
- A product, such as a set of transformations useful for similar data sets.
- One or more ad-hoc exploratory projects.
- An insight, such as a conclusion that you illustrate with several saved visualizations and share with your team.
- A long-term BDD application for further analysis of future data sets.
- An infrastructure (a set of tools that includes Big Data Discovery as a component).

As an outcome, you:

- reach insights for action
- gain understanding for building more sophisticated mathematical models
- create data products, such as new features and services for your organization.

About Studio

Studio is a component in Big Data Discovery. Studio provides a business-user friendly user interface for a range of operations on data.

It combines search, guided navigation, and many data visualizations in a single environment.

Some aspects of what you see in Studio always appear. For example, Studio always includes search, **Explore**, **Transform**, and **Discover** areas. You can add other parts of your interface as needed. These include many types of data visualization components. For example, you can add charts, maps, pivot tables, summarization bars, timelines, and other components. You can also create custom visualization components.

Studio provides tools for loading, exploration, updating, and transforming data sets. It lets you create projects with one or more data sets that are linked. You can load more data into existing projects. This increases the corpus of analyzed data from a sample to a fully loaded set. You can also update data sets. You can transform data with simple and more complex transforms, and write transformation scripts.

Studio's administrators can control data set access and access to projects. They can set up user roles, and configure other Studio settings.

For more information, see Taking a Tour of Studio.

About Data Processing

The **Data Processing** component of BDD runs a set of processes and jobs. This set is called **data processing workflows**. Many of these processes run natively in Hadoop.

The Data Processing component controls several workflows in BDD. For example, you can create workflows for data loading, data updates, and others. The Data Processing component discovers new Hive tables and loads data into BDD. It runs data refresh operations and incremental updates. It also keeps BDD data sets in synch with Hive tables that BDD creates.

For example, during a data loading workflow, the Data Processing component performs these tasks:

- Discovery of data in Hive tables
- Creation of data sets in BDD
- Running a select set of enrichments on discovered data sets
- Profiling of the data sets
- Indexing of the data sets, by streaming data to the Dgraph.

To launch a data processing workflow when Big Data Discovery starts, you use the **Data Processing Command Line Interface (DP CLI)**.

Data Processing CLI

The DP CLI is a shell Linux utility that launches data processing workflows in Hadoop. You can control their steps and behavior. You can run the DP CLI manually or from a `cron` job. You can run the data processing workflows on an individual Hive table, all tables within a Hive database, or all tables within Hive. This depends on DP CLI settings, such as a blacklist and a whitelist.

Here are some of the jobs you can run with DP CLI:

- Load data from Hive tables after installing Big Data Discovery (BDD). When you first install BDD, your existing Hive tables are not processed. You must use the DP CLI to launch a data processing operation on your tables.
- Run data updates. They include:
  - An operation to refresh data. It reloads an existing data set in a Studio project, replacing the contents of the data set with the latest data from Hive in its entirety.
  - An incremental update. It adds newer data to existing data sets in a Studio's project.

- Launch the BDD Hive Table Detector (this is a utility in Data Processing). The BDD Hive Table Detector detects if a new table is added to Hive. It then checks the whitelist and blacklist. If the table passes them, it creates a data set in BDD. It also deletes any BDD data set that does not have a corresponding source Hive table. This keeps BDD data sets in synch with data sets in Hive. For detailed information on how data sets are managed in BDD, see Data set lifecycle in Studio.

For information on Data Processing and the DP CLI, see the *Data Processing Guide*.

About the Dgraph

The Dgraph uses data structures and algorithms to issue real-time responses to queries.

The Dgraph stores the ingested data in the Dgraph databases it creates for the indexed data sets. The Dgraph receives client requests from Studio, queries its databases, and returns the results. The Dgraph is stateless. This design requires that a complete query is sent to it for each request. The stateless design facilitates the addition of Dgraph instances for load balancing and redundancy. Any instance of a Dgraph can reply to queries independently of other instances.

A Big Data Discovery cluster can include two or more Dgraph instances. They handle end-user query requests and store Dgraph databases on shared storage. For each data set, the leader Dgraph (once it is automatically appointed in BDD) handles all write operations, such as updates and configuration changes to that data set. The remaining Dgraphs serve as read-only followers for that data set. The Dgraph Gateway performs the routing of requests to the Dgraph nodes. It handles query caching, business logic, and cluster services for the Dgraph nodes.

The deployment of Dgraphs occurs at the installation time. For information, see the *Installation Guide*.

The Dgraph Tracing Utility

The Dgraph Tracing Utility is a Dgraph diagnostic program used by Oracle Support. It stores the Dgraph trace data, which are useful in troubleshooting the Dgraph. It starts when the Dgraph starts and keeps track of all Dgraph operations. It stops when the Dgraph shuts down. You can save and download trace data to share it with Oracle Support.

The bdd-admin script

Use the `bdd-admin` script to run administrative operations on the BDD cluster deployment, such as starting components, backing up, restoring, and updating the configuration. For information on the `bdd-admin` script, see the *Administrator's Guide*.

The Dgraph HDFS Agent

The Dgraph HDFS Agent loads data into the Dgraph. It reads the Avro files from the data processing workflow and formats them for data loading. It then sends them to the Dgraph. For information on the Dgraph HDFS Agent, see the *Data Processing Guide*.

4 Taking a Tour of Studio

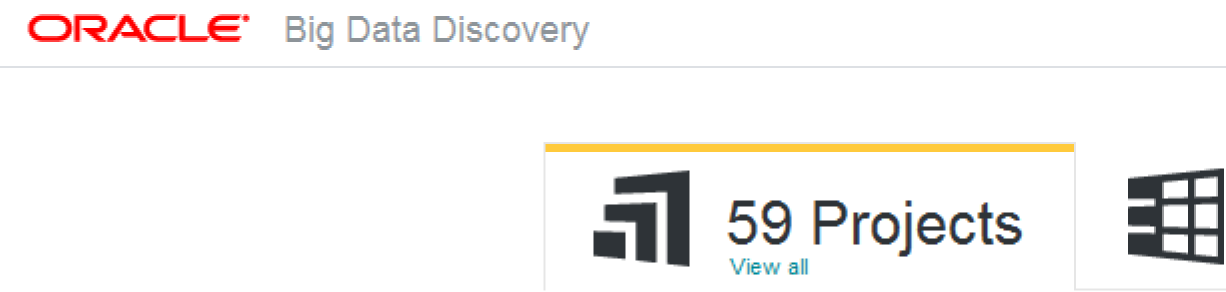
This section walks you through Studio. It describes how you move through areas within Studio when working with big data.

Workflow in Studio

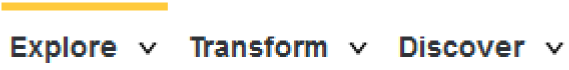
Studio is the visual face of Big Data Discovery. It enables exploratory data analysis and is structured around **Explore**, **Transform** and **Discover** areas.

- After you log in, you see the Catalog. It lets you browse and discover the contents of the data lake. (A **data lake** is a storage repository that holds raw data in its native format until it is needed.)

In Catalog you can find projects you created before, or projects shared by other users. You can also browse data sets discovered in a Hive data base, or data sets that other users uploaded from a file or imported from an external JDBC data source:



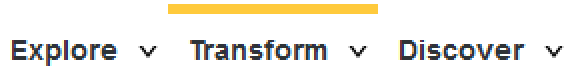
- Next, if there are no **Discover** pages created yet by other users, the **Explore** page for the first data set opens. If **Discover** pages already have been created by other users, the first **Discover** page opens.
- If you move to **Explore**, you can pick data sets that are of interest to you for further exploration, and use the results of data profiling and enrichments to grasp basic data characteristics. For example, you can look for outliers. Outliers are values that are distant from other values for the attribute. You can also explore scatter plots to see correlations between values for two attributes, or link data sets.



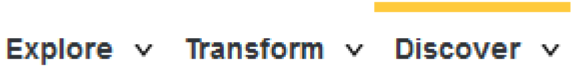
- If some pages by other users have already been created and you have access to them, then instead of **Explore**, you move to **Discover** for the project that is shared with you.
- If you started with **Explore**, then to move to **Transform**, or **Discover**, you must first add the data set to a project. Alternatively, you can find and select an existing project:

Note that once your data set is in a project, you can continue using **Explore**.

- You can then move to **Transform**, where you transform the data and remove inconsistencies. For example, you can change data types, or create custom transformation scripts.



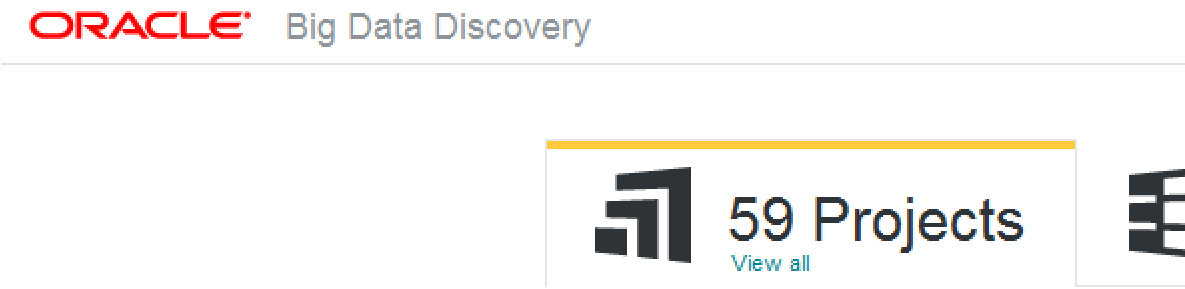
- In **Discover** you arrive at insights about the data and save your projects for sharing with others.



Finding data sets

The number of data sets in any data repository can be huge. In Studio's Catalog, you can use search and Guided Navigation to find data sets and existing projects relevant to your analytical purpose.

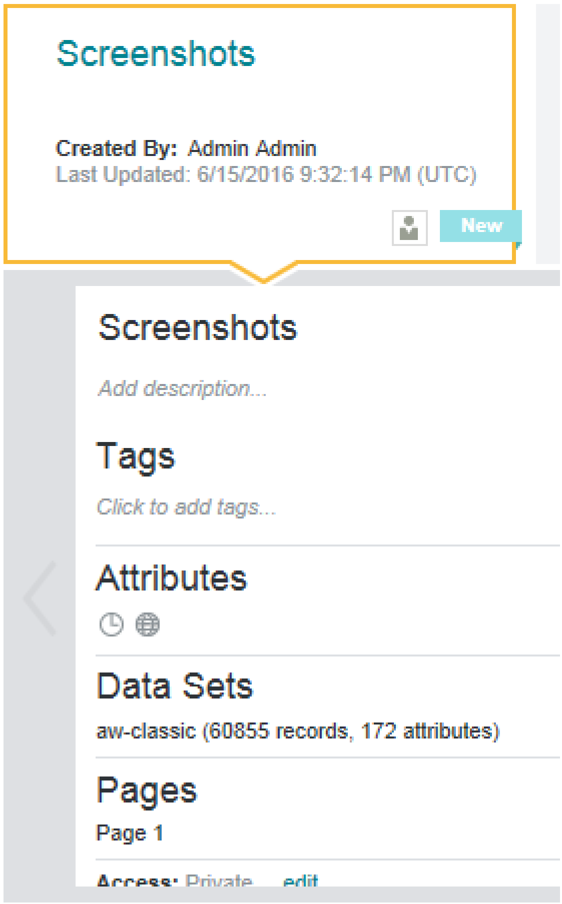
After you log in, you see the **Catalog**:



Here, you can:

- Find data sets
- Find existing projects
- Add a data set to BDD

You can quickly look at a data set by clicking the tile to expand the Data Set Details panel. This displays the data set’s enriched metadata to provide a summary to help you decide if you'd like to explore this data set further:



Screenshots

Created By: Admin Admin
Last Updated: 6/15/2016 9:32:14 PM (UTC)

Screenshots

Add description...

Tags

Click to add tags...

Attributes

Data Sets

aw-classic (60855 records, 172 attributes)

Pages

Page 1

Access: Private edit

Through the Catalog, you can:

- Navigate data sets using a familiar keyword search and Guided Navigation experience similar to that found on major commerce sites.
- Browse data sets based on name, metadata, attribute names, topics, dates, locations, or key terms.
- Find related data sets that are potentially relevant to your interests based on overlapping data or frequent use by other users.
- Filter displayed data sets based on data set metadata (such as the creation date, tags, notes, number of records, or contents of the data) and also by attribute metadata (such as semantic types).

Adding tags and notes to data sets

When you expand a data set to view the data set details, you can optionally add tags or notes that describe the data. All users can then filter data sets based on those tags, or run keyword searches that include text from the data set notes.

To add notes, click the description field below the data set name.

To add searchable tags, click the text field below **Tags**. To help you choose consistent tag names, autocomplete results may display as you type.

Profiling and Enriching Data

This topic summarizes how Big Data Discovery helps to profile and enrich new data.

You can add new data as files in HDFS using data integration technologies. You can also use Studio to upload files, such as Excel and CSV, or pull data from a database using your credentials, and import it into a personal sandbox in HDFS. In either case, when it loads data, Big Data Discovery performs these activities for you:

- Profiles the data, inferring data types, classifying content, and understanding value distributions.
- Lists most interesting data sets first and indicates what they contain.
- Decorates the data with metadata, by adding profile information.
- Enriches the data, by extracting terms, locations, sentiment, and topics, and storing them as new data in HDFS.
- Takes a random sample of the data of the specified size. (You can increase the sample size, or load full data.)
- Indexes the data and prepares it for fast search and analysis in BDD.

As an outcome, the data you need resides in HDFS, is indexed by the Dgraph and enriched by BDD, and is ready for your inspection and analysis.

Using Explore

Explore provides you with an out-of-the-box guided analytics experience. It configures the canvas with the right visualizations to explain the data based on your goals. This allows you to get a summary of the data and explore data quality.

Explore provides an attribute-focused visual summary of the data, summarizing value distributions, data quality gaps, and relationships.

Explore presents visualizations that give you the most insight into the data set, selecting the types most suitable to each attribute's data type and value distribution.

Visualizations are automatically composed, to save you time and effort at this early stage in the process. When you have a better understanding of the data set, you can compose your own visualizations on data that you have identified as worthy of further analysis.

You can think of the **Explore** area as a guided tour of new data sets, freeing you from the need to manually query the data or configure your own visualizations. It helps you immediately extract meaning from the results to better understand what's inside the data set.

Here are some of the questions that Big Data Discovery provides answers to, when you use **Explore**:

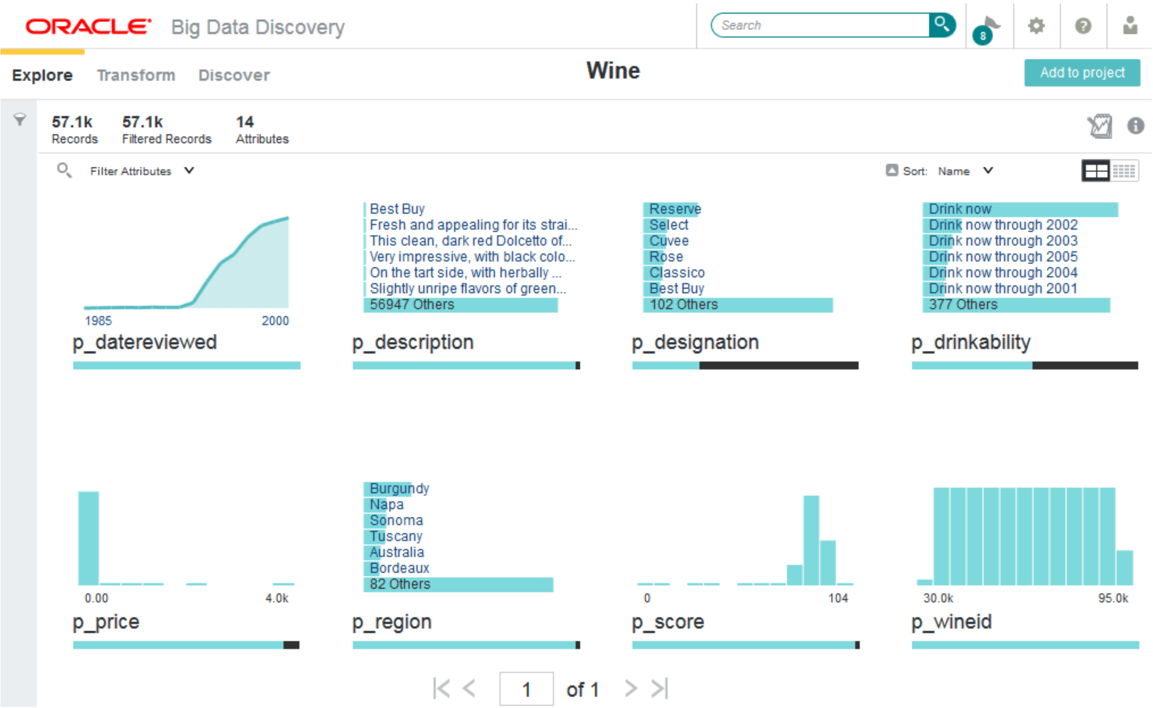
- What fields are in my data? Do I understand what they are? **Explore** shows these fields as a series of visualizations.
- How are values distributed? **Explore** uses descriptive statistics (mean, median, mode, quintiles) and visuals (histograms, box plots).

- How dirty/messy is the data? Are there missing values or outliers? **Explore** uses the bar underneath each attribute, showing you at a glance whether any inconsistencies exist.
- What relationships exist between fields? **Explore** uses numeric correlations and visuals.

In addition, **Explore** lets you add your own notes and tags to data sets. You can later sort or search on those tags, or review your notes for a quick summary without having to walk through the data a second time.

As an outcome, you understand the data sets you've been exploring.

Here are some examples of **Explore** visualizations:



This image shows **Explore** for a data set with 57.1K records, with attributes sorted by name.

Note: Even if you could explore a large data set at full scale, you always want to start by exploring a representative sample, and later confirm your hypotheses or expand your analysis at full data scale.

Using Transform

Through exploration, data quality issues often come to the surface. You must triage them and, in many cases, correct, clean, or change data in some way. The interactive **Transform** area in Studio helps you turn your data into data that is fit for advanced analysis.

Transform helps you isolate data quality problems and lets you quickly apply a number of data transformations to clean or restructure the data.

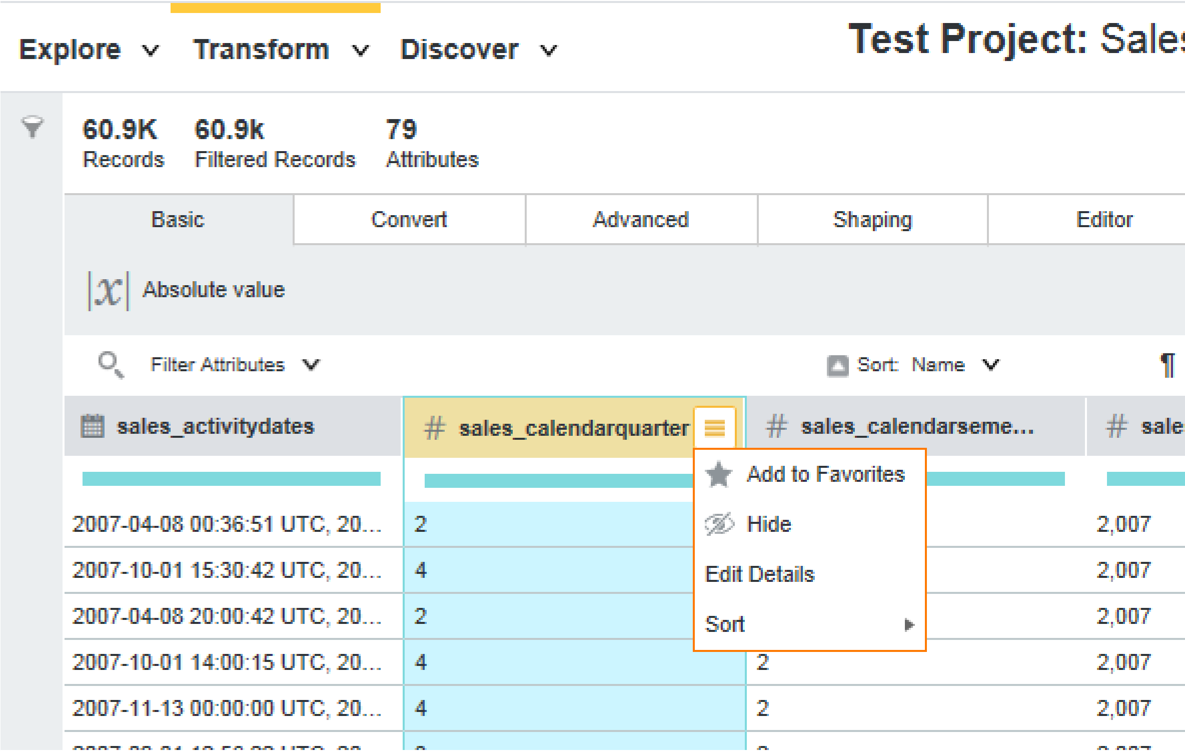
You can find **Transform** in Studio here:

Transform presents a spreadsheet-style view of data, organized either by attribute's value list view, or row view. Here are examples of how you can transform data:

- You can focus on just a few basic transformations needed to support your visualization or analysis. You do this by running common data transformations, such as filling missing values, converting data types, collapsing values, reshaping, pivoting, grouping, merging, and extracting.
- You can join one or more data sets together within a project and also aggregate attribute values to create new derived attributes that sum, average, find minimum/maximum values, and run other operations.
- You can use live **Preview** to see the set of transformations in action before applying them to the entire data set in Studio.

  Because data transformation and data exploration are two integral aspects of any discovery project, you can seamlessly transition between **Transform** and **Explore** within a unified interface, which promotes a smooth workflow.
- You can also share your transformation scripts with other BDD users in your team.

Here is an example of what **Transform** looks like in a project:



sales_activitydates	sales_calendarquarter	sales_calendarseme...	sale
2007-04-08 00:36:51 UTC, 20...	2		2,007
2007-10-01 15:30:42 UTC, 20...	4		2,007
2007-04-08 20:00:42 UTC, 20...	2		2,007
2007-10-01 14:00:15 UTC, 20...	4	2	2,007
2007-11-13 00:00:00 UTC, 20...	4	2	2,007

In this image, you can see the **Transform** area of Studio. Notice the drop-down attribute menu in the second column. It includes options to hide an attribute, mark it as favorite, edit it, and sort it. To the right, you can also see the **Transform Script** panel, with three transforms created by the project's user.

Also notice the transform menu just under the header. The Transform menu is made up of the tabs: **Basic**, **Convert**, **Advanced**, **Shaping**, and **Editor**.

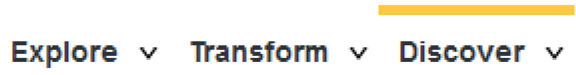
The transforms that display under each tab vary depending on the data type of the attribute you select. For example, if you select a date time attribute, the **Basic** tab displays the **Truncate date** and **Extract date part** transforms. These are both transforms that are contextually appropriate for date time attributes. And if you select a numeric attribute, the **Basic** tab displays the **Absolute value** transform because the **Absolute value** transform is contextually appropriate for numeric attributes.

After you run the transform script by clicking **Commit to Project**, the data is transformed and ready for further analysis.

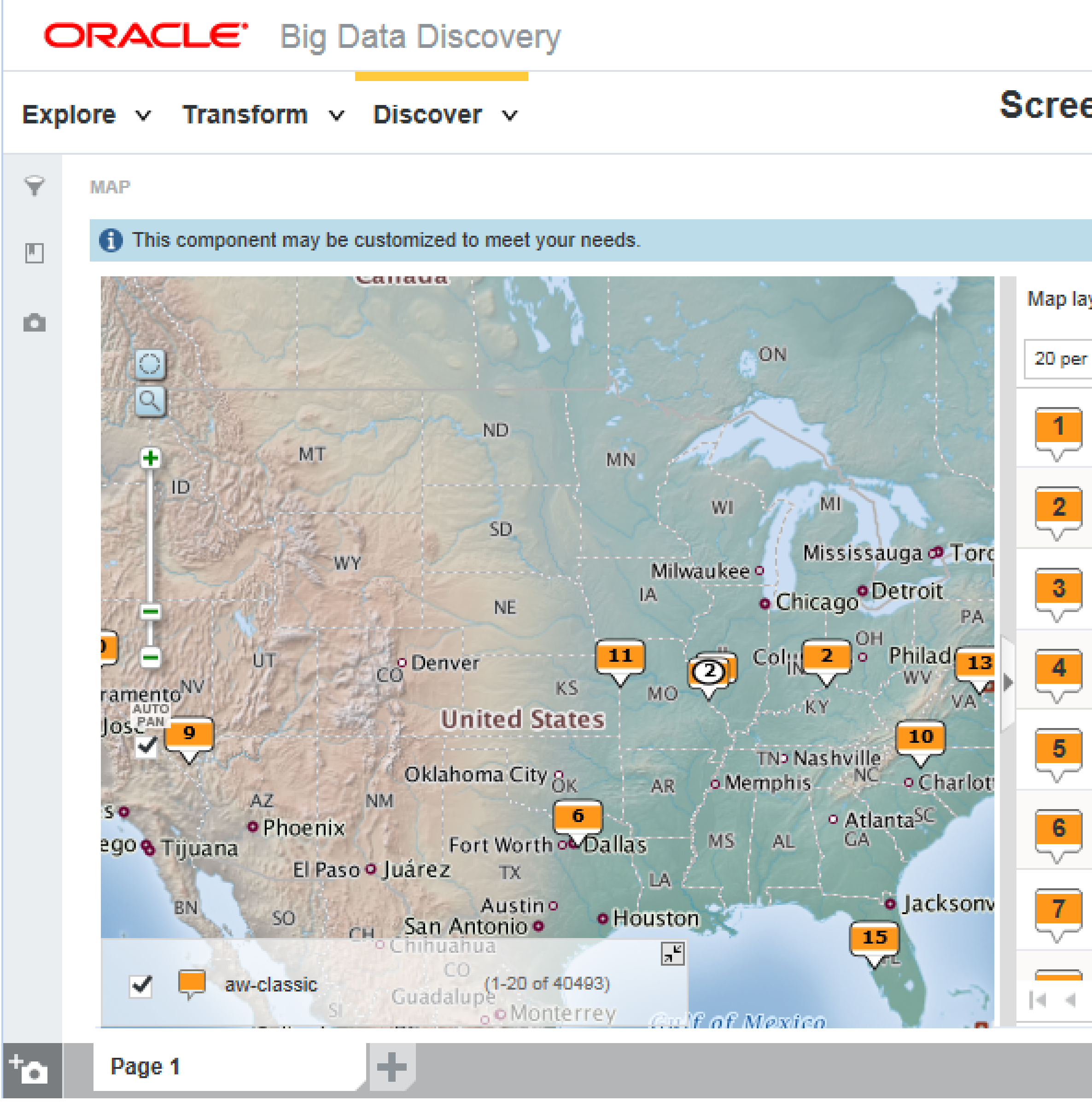
Using Discover

Using **Discover** in Studio, you analyze the data and discover insights hidden in data. **Discover** lets you identify and pick meaningful visualizations and capture the results of the data exploration phase.

You can find the **Discover** area of Studio here:



Here is an example of how **Discover** might look for your project:



In this diagram, you can see the **Discover** area, with the Map visualization added. To the right, notice some of the other visualization components you can add to your project.

Some of the examples of the insights you can create with **Discover** are:

- Use search and Guided Navigation to identify subsets of data for deeper analysis.
- Use interactive visualizations to let the data speak for itself, in raw or aggregated forms, in text or in numbers. Statistical techniques (regression, correlation) and machine learning tools (classification, clustering) are used in Big Data Discovery to drive these visualizations.
- Join (or combine) multiple data sets for new insights.
- Create projects that serve as models for different outcomes. These models automatically understand the weight of different factors in contributing to given outcomes without requiring experts to hand-code rules or heuristics.

- Add new data to existing projects or run your projects on existing data that has been altered to examine new scenarios. (This is known as "what if" analysis.)
- Export your data sets from Big Data Discovery back into Hadoop, so that you can run complex models built in other tools (such as R, or Python) on your custom-created data sub-sets, such as on different customer segments.

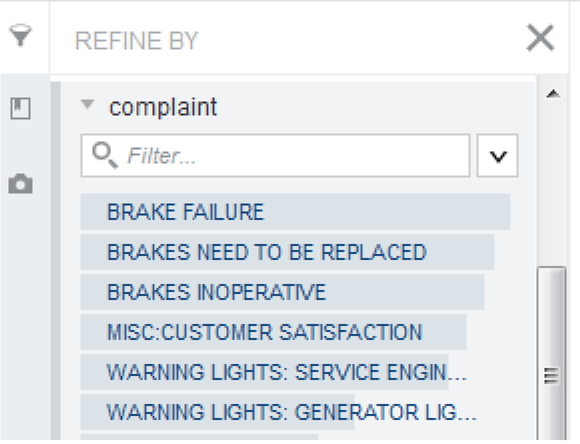
Using filtering, Guided Navigation, and Search

The following examples explain how search filtering and Guided Navigation work in Big Data Discovery.

The Catalog, **Explore**, **Transform**, and **Discover** areas in Studio make use of powerful search and data-driven Guided Navigation capabilities of Big Data Discovery.

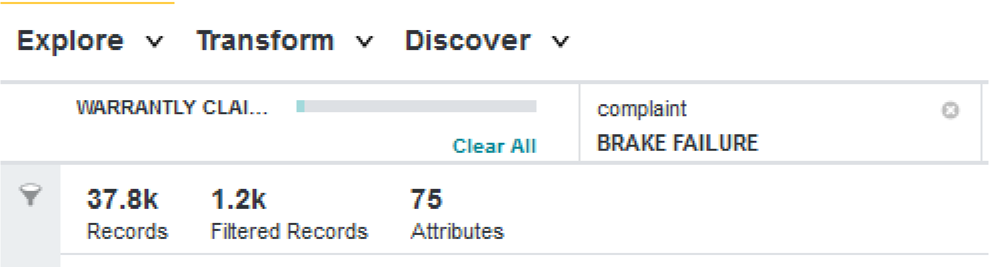
In the Catalog, refinements are always available on the left sidebar.

For data sets in projects, you can expand or collapse the sidebar using the funnel icon. If you expand it, it looks like this:


REFINE BY
complaint
Filter...
BRAKE FAILURE
BRAKES NEED TO BE REPLACED
BRAKES INOPERATIVE
MISC:CUSTOMER SATISFACTION
WARNING LIGHTS: SERVICE ENGIN...
WARNING LIGHTS: GENERATOR LIG...

Using refinements, you can filter on attributes or on available metadata such as creation date. In the Catalog, this also includes any tags that users add to projects or data sets.

Selected refinements represent applied filters. They show up as breadcrumbs at the top of your screen. Here is an example of a selected refinement from the attribute `complaint = BRAKE FAILURE`. Notice that applying the refinement reduces the number of matching records to only 1.2k of the 37.8k total records in the data set.



Attribute filters apply across all parts of BDD. If you highlight or exclude attributes, you project reflects these selections when you move from the Catalog to **Explore** to **Transform** and back. For example, if you select all numeric attributes in a data set in **Explore**, and then switch to **Transform**, **Transform** is filtered by all numeric attributes.

You can also search data sets at any stage in your workflow using the search box:

In addition to searching across project and data set content, keyword search results from the Catalog also include any notes that users add to describe projects or data sets.

Like refinement filters, keyword search filters apply across both **Explore** and **Transform** if you switch between pages.

5

Data Sets in Big Data Discovery

This section describes what is a data set in BDD, discusses sampled and full data sets, and access to data sets. It also discusses a data set's lifecycle in Studio, and BDD projects and applications.

About data sets

Data sets in BDD are called **BDD data sets**. This helps you to distinguish them from **source data sets** in Hive tables.

A BDD data set is the central concept in the product architecture. Data sets in BDD originate from these sources:

- Many data sets are loaded as a result of the data processing workflow for loading data. It runs when you launch DP CLI, after installing Big Data Discovery. This process adds data sets to Studio's Catalog.
- Other data sets appear in Studio because you load them from a personal file or a JDBC data source.

- Also, you can create a new BDD data set by transforming an existing data set, or by exporting a data set to HDFS as a new Hive table. Such data sets are called **derived data sets**.

When the Data Processing component runs data loading, or when you add a data set by uploading a file or data from a JDBC source, it appears in Studio's Catalog. You can use **Preview** to see the details for all data sets in Catalog.

You can also use Studio's **Data Set Manager** to see information about all data sets in a project. For each data set in a project, you can see its record count, when it was added, whether it is private to you, and other details.

Data sets in Catalog vs data sets in projects

It is important to distinguish between data sets in the Catalog, and data sets you add to a project.

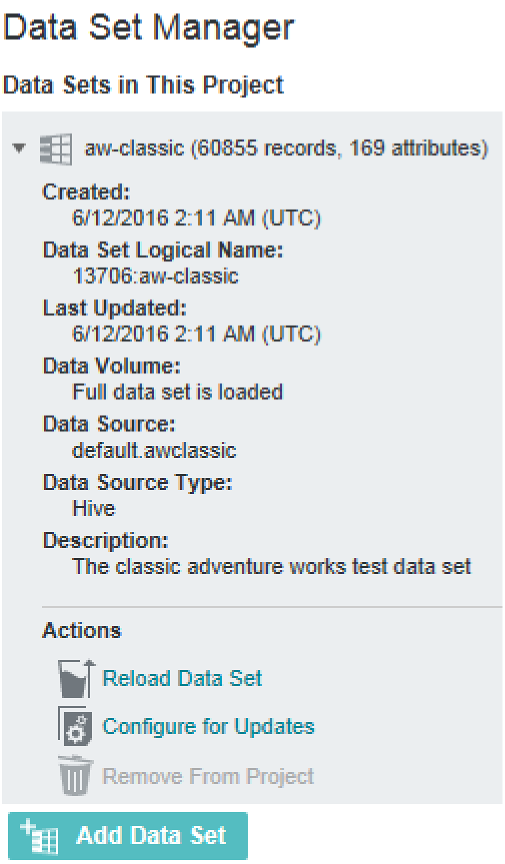
Here is why:

- The Catalog only contains data sets that are discovered in a data lake, or that you add to BDD from a personal file or import from a JDBC source. A data set in a project represents your own modified version of the data set. Once you move a data set to a project, it is similar to creating a personal branch version of the data set.
- You can edit data set metadata and some attribute metadata from the Catalog or the **Explore** view, but data set altering operations, such as transformations, enrichments, and other attribute metadata changes, require editing the data set in a project.
- You can perform some updates of data sets from the Catalog, such as reloading a newer version of a data set. In order to run scheduled updates using the DP CLI, a data set must be part of a project. This is beneficial if you want to run updates periodically, and automate them, by adding them to your scripts that use DP CLI.

Data Set Manager

In Studio, the **Data Set Manager** includes information about all project data sets. You access Data Set Manager once you are in a project, from **Project Settings**.

Here is an example of data set's details in the Data Set Manager:



Data Set Manager

Data Sets in This Project

aw-classic (60855 records, 169 attributes)

Created:
6/12/2016 2:11 AM (UTC)

Data Set Logical Name:
13706:aw-classic

Last Updated:
6/12/2016 2:11 AM (UTC)

Data Volume:
Full data set is loaded

Data Source:
default.awclassic

Data Source Type:
Hive

Description:
The classic adventure works test data set

You can see that the project includes one data set. You can also see the number of records and attributes in it.

The **Data Source Type** shows the type of the source file, such as Excel, or CSV, or it shows "Hive", if the data originated from a Hive table. In this example, the Data Source Type is Excel. This means the data set originated from the spreadsheet that was loaded in Studio and then was reloaded. You can tell this by observing the dates for creation and update.

The **Data Source** field shows the name of the source file. In this example, it is `WarrantyClaims.xls`.

Notice the **Data Set Logical Name**. This is the name that each data set has, whether it exists in Catalog or belongs to a project. When a data set is in Catalog, it has one data set logical name. When you move it into a project, the data set's logical name changes. To run scripted updates with DP CLI, you need to note the correct data set logical name, so that you know which data set you are going to update. For information on scripted updates, see the *Data Processing Guide*.

Sampled and full data sets

Data sets in BDD can be sampled, or they can represent a full data set.

Sampled data sets

A *sample data set* in BDD represents a random sample of a source data set in Hive. If the data set originates in Hive, you use the Data Processing CLI to load it. The DP CLI uses the default sample size of 1 million records. You can specify a different sample size at data loading time.

- If you specify a sample size that is less than the size of the source data set in Hive, a sample data set is loaded into BDD.
- If you specify a sample size that is greater than or equal to the size of the source data set, a full data set is loaded.

Full data sets

A *full data set* in BDD represents a data set that contains all records, if you compare it to the source it was loaded from. For example, if a data set originates in Hive, and the sample size in DP CLI is greater than the record count in the source Hive table, this data set is loaded in full.

For a summary of how to get from a sample to a full data set, see Data loading and sample size

For more information on sampling and data set loading during data processing, see the *Data Processing Guide*.

For information on adding and managing data sets in Studio, including loading a full data set, see the *Studio User's Guide*.

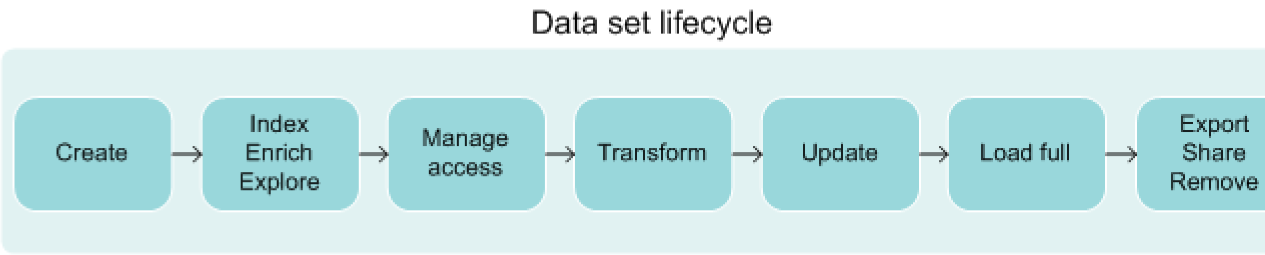
Data set lifecycle in Studio

As a data set flows through Big Data Discovery, it is useful to know what happens to it along the way.

Before we describe the data set lifecycle, here's how BDD interacts with source data sets it finds in Hive:

- BDD does not update or delete source Hive tables. When BDD runs, it only creates new Hive tables to represent BDD data sets. This way, your source Hive tables remain intact if you want to work with them outside of Big Data Discovery.
- Most of the actions in the BDD data set lifecycle take place because you select them. You control which actions you want to run. Indexing in BDD is a step that runs automatically.

This diagram shows stages in the data set's lifecycle as it flows through BDD:



In this diagram, the data set goes through these stages:

1. **Create** the data set. You create a data set in BDD in one of two ways:
   - Uploading source data using Studio. You can upload source data in delimited files and upload from a JDBC data source. When you upload source data, BDD creates a corresponding Hive source table based on the source data file.
   - Running the Data Processing CLI to discover Hive tables and create data sets in Studio based on source Hive tables. Each source Hive table has a corresponding data set in Studio.

The Catalog of data sets in Studio displays two types of data sets. Some data sets originate from personally-loaded files or a JDBC source. Other data sets are loaded by Data Processing from source Hive tables.

2. Optionally, you can choose to **enrich** the data set. The data enrichment step in a data processing workflow samples the data set and runs the data enrichment modules against it. For example, it can run the following enrichment modules: Language Detection, Term Frequency/Inverse Document Frequency (TF/IDF), Geocoding Address, Geocoding IP, and Reverse Geotagger. The results of the data enrichment process are stored in the data set in Studio and not in the Hive tables.

   Note: The Data Processing (DP) component of BDD optionally performs this step as part of creating the data set.

3. **Create an index** of the data set. The Dgraph process creates binary files, called the Dgraph database, that represent the data set (and other configuration). The Dgraph accesses its database for each data set, to respond to Studio queries. You can now explore the data set.

4. **Manage** access to a data set. If you uploaded a data set, you have private access to it. You can change it to give access to other users. Data sets that originated from Hive are public. Studio's administrators can change these settings.

5. **Transform** the data set. To do this, you use various transformation options in **Transform**. In addition, you can create a new data set (this creates a new Hive table), and commit a transform script to modify an existing data set.

   If you commit a transform script's changes, Studio writes the changes to the Dgraph, and stores the changes in the Dgraph's database for the data set. Studio does not create a new Hive table for the data set. You are modifying the data set in the Dgraph, but not the source Hive table itself.

6. **Update** the data set. To update the data set, you have several options. For example, if you loaded the data set from a personal data file or imported from a JDBC source, you can reload a newer version of this data set in Catalog. If the data set was loaded from Hive, you can use DP CLI to refresh data in the data set.

   Also, you can **Load Full Data Set**. This option is useful for data sets that represent a sample. If a data set is in a project, you can also configure the data set for incremental updates with DP CLI.

7. **Export** the data set. When your data set is in a project, you can export it. For example, you can export a data set to HDFS, after you have applied transformations to it. This way, you can continue working with this data set using other tools. Also, you can export a data set and create a new data set in Catalog. Even though on this diagram **Export** is shown as the last step, you can export the data set at any stage in its lifecycle, after you add the data set to a project.

8. **Share** the data set. At any stage in the data set's lifecycle, you can share the data set with others.

9. **Remove** the data set. When you delete a data set from Studio, the data set is removed from the Catalog and is no longer accessible in Studio. However, deleting the data set does not remove the corresponding source Hive table that BDD created, when it loaded this data set.

It's important to note that BDD does not update or delete original source Hive tables. BDD only creates new Hive tables to represent BDD data sets. You may need to ask your Hive data base administrator to remove old tables as necessary to keep the Hive data base clean. If the Hive data base administrator deletes a Hive table from the data base, the Hive Table Detector detects that the table was deleted and removes the corresponding data set from the Catalog in Studio. The Hive Table Detector is a utility in the Data Processing component of BDD.

Projects and applications in Big Data Discovery

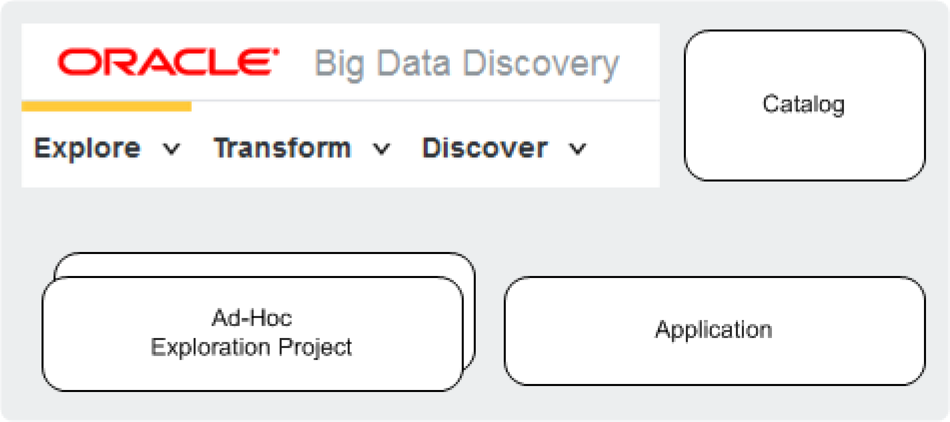
When you work with data sets in BDD, you add them in projects in Studio. Some BDD projects can become BDD applications.

BDD projects let you perform ad-hoc exploration and discovery using standard and advanced analytic techniques.

BDD applications expose an interactive analytic dashboard to a broad set of users.

While each BDD application is also a BDD project in Studio, it is referred to as a BDD application because it has a number of characteristics that make it special. In other words, you always start with projects in Studio; you can turn some projects into BDD applications.

The following diagram illustrates that BDD can contain one or more projects and an application:



In the diagram, the Catalog is also shown. It contains data sets that you select to add to your projects. Some of the projects you can later turn into BDD applications.

BDD projects

BDD projects are created by each user and serve as personal sandboxes. Each BDD deployment supports many BDD projects at once. This lets everyone in the BDD analyst community explore their own data, try different sample data sets, and identify interesting data sets for future in-depth analysis.

BDD projects often, but not always, run on sample data and allow you to load newer versions of sample data into them. Each BDD deployment can support dozens of ad-hoc, exploratory BDD projects. You can turn the most interesting or popular BDD projects into BDD applications.

Here are characteristics of a BDD project. It:

- is often a short-lived project, used to try out an idea

- is owned by a single user who can maintain, edit, and delete the project once no longer needed
- typically, but not always, runs on a sample of data
- answers a simple analytics question, or helps with initial exploration of the data set in Catalog
- lets you identify and select data set candidates for in-depth analysis.

As opposed to BDD projects that any user in BDD can create, BDD administrators own and certify BDD analytic applications, which they can share with their teams.

BDD applications

A **BDD application** includes a list of data sets that have been tweaked, linked and transformed, based on the goals of business analysis. Data analysts with power user permissions create and configure it. They can share it with other users in the BDD business analyst community who can use it for analysis.

Such an application answers a set of predefined questions and lets a group of users analyze the findings in it. A BDD application is often built on at least one full data set, with other data sets linking to it.

It is configured for periodic scripted data updates. You can either load newer versions of data onto it, or keep the existing data while periodically adding new data.

Each BDD deployment can support several BDD applications. The number of applications a BDD deployment can support depends on the capacity and sizing of your BDD deployment.

You can think of a BDD application as a "productized" or "certified" project that gained greater significance and is now maintained by power users (or by your IT organization) for use by a group of analysts within your team.

Here are the characteristics of a BDD application. It:

- is a longer-term project as opposed to other BDD projects
- often includes joins from multiple data sets
- often contains full data from at least one of the comprising data sets
- includes a predefined set of visualizations aimed at answering a set of specific questions
- allows you to run periodic data updates implemented with DP CLI
- has user access for a BDD application that is split between owners, curators and the team. Owners create and configure it, curators maintain it; the team can view and use visualizations. The team members may have different access to data sets in the application.
- has results and findings that are intended for sharing and viewing by a team (as opposed to exploratory BDD projects used by individual users who create them).

Data set access, access to projects, and user roles

Studio Administrators and project authors manage access to data sets and projects in Studio.

User roles

In Studio, two sets of user roles exist: a set of global permissions, and project-specific permission sets. For information on Studio's project access, data set access and user roles, see the *Administrator's Guide*.

Data set access

If you have Read access to a data set, you can see it in search results, or by browsing the Catalog. You can also explore it or add it to a project and then modify the project-specific version of the data set.

Write access allows you to set data set metadata or change data set permissions.

For information on changing access to a data set, see the *Studio User's Guide*.

For information on changing the default Studio settings for data set access, see the *Administrator's Guide*.

Project access

Projects are private by default. Only the project author and the Studio Administrators group can access them. If you are an Administrator or the project author, you can modify access to add user groups or individual users to your project. For information on changing access to a project, see the *Studio User's Guide*.

6

Data Loading and Updates

This section discusses options for initial data loading and data updates. It illustrates how you can load files in Studio, or using Data Processing CLI.

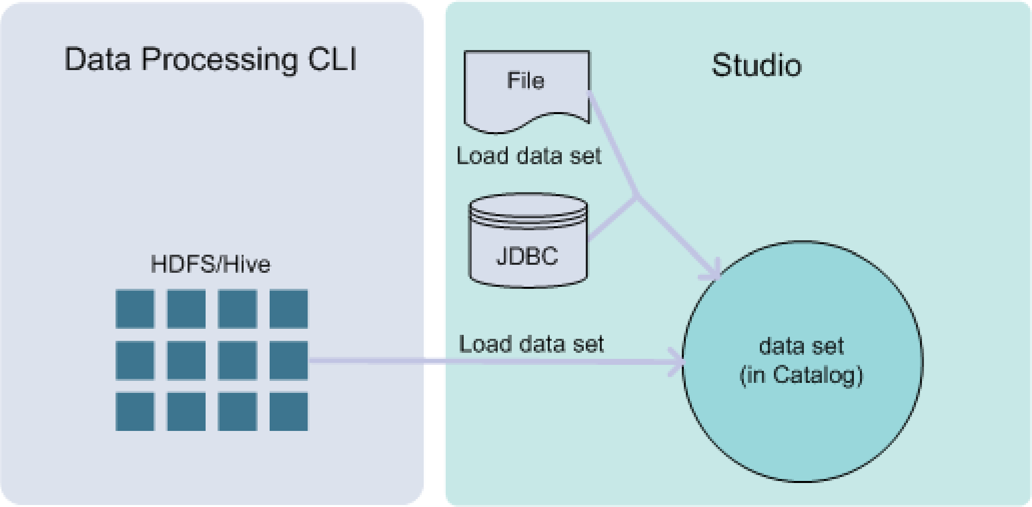
Data loading options

BDD offers several options for data loading. You can load data by running the data loading workflow with DP CLI. Also, in Studio, you can upload a personal file or import data from a JDBC source.

For initial loading of data into BDD, three methods exist:

1. **Load of Hive tables**. When you run DP CLI, it runs the data processing workflow and loads data from Hive tables into BDD.
2. **Personal file upload**. In Studio you can upload a data set from a personal file.
3. **Import from a JDBC source**. In Studio you can import a data set from a JDBC source.

This diagram illustrates data loading options:



Also, you can load a sample or full data. Or, you can also start with a sample and then load full data. See Data loading and sample size.

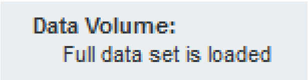
Data loading and sample size

You can load either a sample or a full data set. If you load a sample, you can go to a full data set later. This topic summarizes how to get from a sample to a full data set.

These options are high-level summaries only. For detailed steps, see the referenced documentation for each option.

- **Controlling sample size when loading from Data Processing CLI**. DP CLI has a parameter for data size sample. The default sample size is 1 million records. When you use DP CLI for data loading, you can customize this parameter:

  – If it is less than the record count of source records in Hive, a full data set is loaded. In this case you already loaded a full data set. This is indicated in the Data Set Manager in Studio, in the **Data Volume** field:

  Data Volume:
  Full data set is loaded

  – If it is greater than the number of records in Hive, a sampled data set is loaded, based on the sample size you specify. In this case, you can use DP CLI with `--Incremental update` flag, or you can use **Load Full Data Set** in Studio, to load the entire source data set from Hive. You will then have a full data set in BDD.

  For detailed information on specifying the sample size with DP CLI, see the *Data Processing Guide*.

- **Controlling data set size when loading from a file or a JDBC source**.

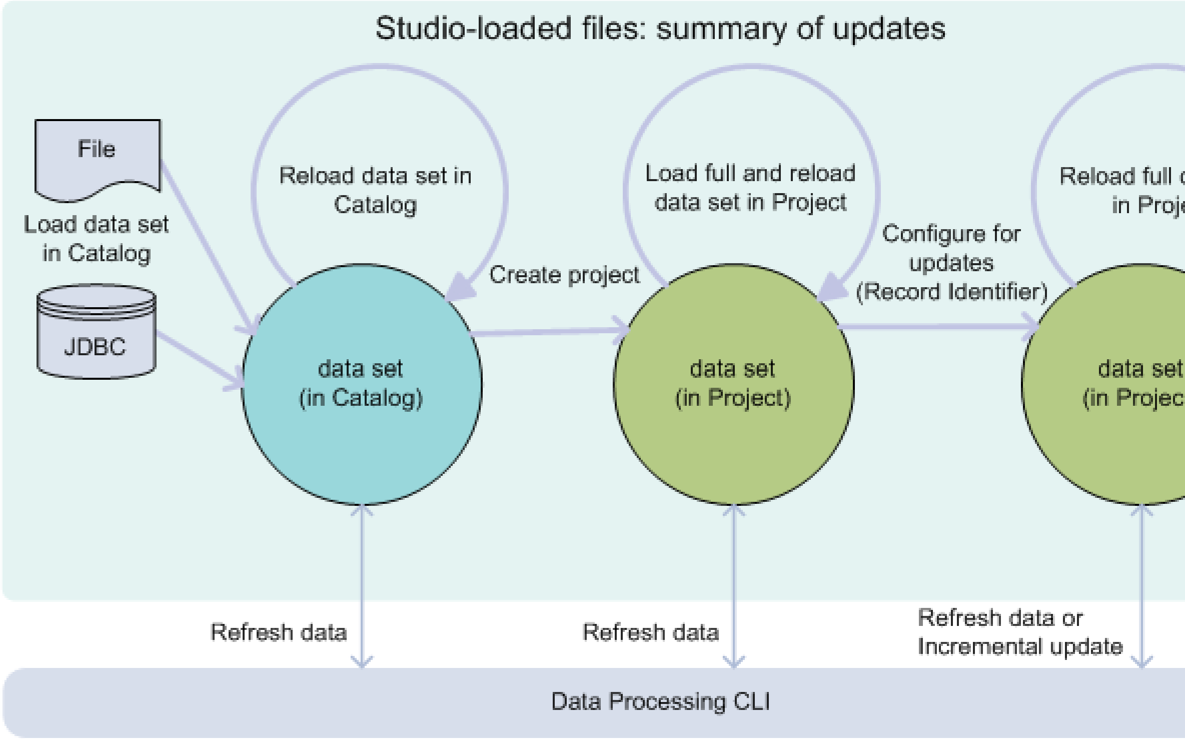
  If you load a data set from a personal file or import it from a JDBC source, then all data is loaded. However, it may still be a sample if you compare it to the source data you may also have elsewhere on your system.

  If you later want to add full data from the source, you can locate the Hive data set that BDD created when you loaded a file. Next, use the `drop` command to place that data set in Hue, and replace it with a production Hive table. You can then run **Load Full Data Set** on this table in Studio. This will load a full data set.

This process is known as creating a BDD application. For detailed steps on this procedure, see the topic on creating a BDD application in the *Studio User's Guide*.

Studio-loaded files: data update diagram

The diagram in this topic shows data sets loaded in Studio by uploading a personal file or importing data from a JDBC source. It illustrates how you can reload this data set in Studio. Also, you can update the data set with DP CLI, and increase its size from sample to full.


Studio-loaded files: summary of updates
File
Load data set in Catalog
JDBC
Reload data set in Catalog
data set (in Catalog)
Create project
Load full and reload data set in Project
data set (in Project)
Configure for updates (Record Identifier)
Reload full d
in Proje
data set
(in Projec
Refresh data
Refresh data
Refresh data or Incremental update
Data Processing CLI

In this diagram, the following actions take place, from left to right:

- You load data with the Studio's **Add data set** action. This lets you load a personal file, or data from a JDBC data source. Once loaded, the data sets appear in Catalog. BDD creates a Hive table for each data set. This Hive table is associated with the data set in Catalog.
- If you later have a newer version of these files, you can reload them with **Reload data set** action, found in data set's details, in Catalog.
- Next, if you'd like to transform, or to use **Discover** area to create visualizations, you should create a project. This is shown in the diagram in the middle circle. Notice that in this circle, the data set is now in your project and is no longer in Catalog. This data set can still be shown in Catalog, for others to use, if they have permissions to see it. However, now you have your own "version" of this data set, inside your project.
- Next, you can choose to load full data into this data set with **Load full data**, found in Data Set Manager. This replaces the data set sample with a fully loaded data set within your project. Or, if the data set was already fully loaded, you can reload the

full data set in your project, by using this option again. This is shown as a circular arrow above the middle circle.

- Alternatively, you can use the options from Data Processing CLI to run scripted updates. There are two types of scripted updates: `Refresh Data` and `Incremental update`.
- To run `Refresh Data` with DP CLI, identify the data set's logical name in the properties for this data set in Studio. You must use that data set's logical name as a parameter for the data refresh command in DP CLI.
- To run `Incremental update` with DP CLI, you must specify a primary key (also known as record identifier in Studio). To add a primary key, use **Control Panel** > **Data Set Manager** > **Configure for Updates**. You also need the data set's logical name for the incremental update.
- Be careful to use the correct data set logical name. Notice that a data set in Catalog differs from the data set in a project.
- Notice that you can run **Reload data set** in Catalog only for data sets in Catalog. If you want to update a data set that is already added to a project, you need to use one of the scripted updates from DP CLI.

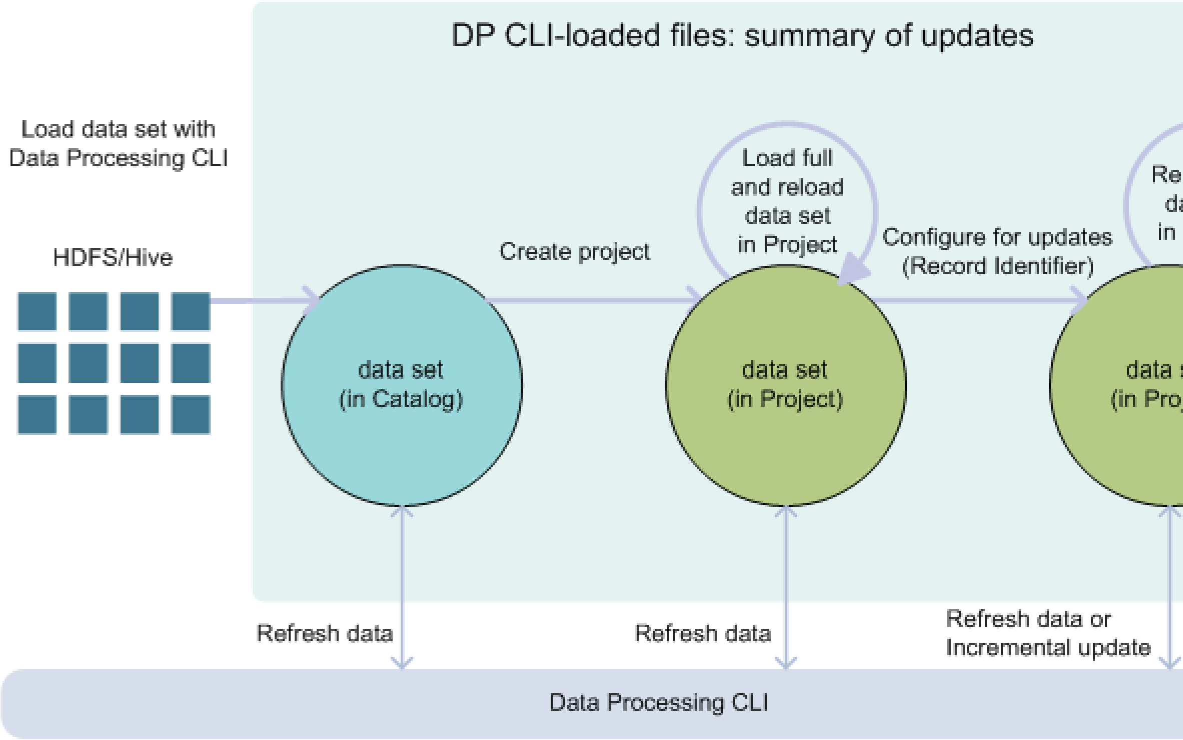
  If you reload a data set in Catalog that has a private copy in a project, Studio alerts you to this when you open the project, and gives you a chance to accept these updates. If any transformations were made to the data set in the project, you can apply them to the data set in Catalog.

Note the following about this diagram:

- You can `Refresh Data` with DP CLI for data sets in Catalog and in projects. Typically, you use DP CLI `Refresh Data` for data sets in a project.
- You can run an `Incremental Update` with DP CLI after you specify a record identifier. For this, you must move the data set into a project in Studio.
- Once you load full data, this does not change the data set that appears in Catalog. Moving a data set to a project is similar to creating your personal version of the data set. Next, you can load full data and write scripted updates to this data set using DP CLI commands `Refresh data` and `Incremental update`. You can run these updates periodically, as `cron` jobs. The updates will run on your personal version of this data set in this project. This way, your version of this data set is independent of the data set's version that appears in Catalog.

DP CLI-loaded files: data update diagram

The diagram in this topic shows data sets loaded by Data Processing component of BDD, from Hive. The diagram illustrates how you can update this data set using DP CLI, and increase its size from sample to full.


DP CLI-loaded files: summary of updates
Load data set with Data Processing CLI
HDFS/Hive
Create project
Load full and reload data set in Project
Configure for updates (Record Identifier)
Re
da
in
data set (in Catalog)
data set (in Project)
data s
(in Pro
Refresh data
Refresh data
Refresh data or Incremental update
Data Processing CLI

In this diagram, from left to right, the following actions take place:

- You load a data set from Hive using the Data Processing workflow (DP CLI). The data set appears in Catalog in Studio.
- You can now create a BDD project from this data set. Notice that a data set in your project does not eliminate this data set in Catalog. That is, other users may still see this data set in Catalog, if they have permissions. However, you have now taken this data set with you into your project. You can think of it as a different version of the data set, or your project's private version of the data set.
- In a project, you can load full data into it with the **Load full data set** action in Studio.
- Using Data Processing CLI, you can also run scripted updates to this data set. Two types of scripted updates exist: `Refresh Data` and `Incremental Update`.

  To run scripted updates, you need a data set logical name, found in the data set's properties in Studio. It is important to provide the correct data set logical name to the DP CLI. If the data set is in Catalog, it is not the same data set that you have in your project. Note the correct data set logical name.

Note the following about this diagram:

- You cannot run an `Incremental Update` with DP CLI for a data set in Catalog; you can only run it when you add the data set to a project.
- You can run an `Incremental Update` with DP CLI after you specify a record identifier. For this, you must move the data set into a project in Studio.
- Because this data set arrived from Hive, you cannot update it within Studio. Instead, you can use DP CLI for data set updates.

- Once you load full data, this does not change the data set that appears in Catalog. Moving a data set to a project and loading full data is similar to creating a personal version of the data set. Next, you can write scripted updates to this data set, using DP CLI commands `Refresh data` and `Incremental update`. You can run these updates periodically, as `cron` jobs. The updates will run on your personal version of this data set in this project. This way, your version of this data set is independent of the data set's version that appears in Catalog.

With this workflow, you create a project of your own, based on this data set, where you can run scripted updates with DP CLI. This approach works well for BDD projects that you want to keep around and populate with newer data.

This way, you can continue using the configuration and visualizations you built in Studio before, and analyze newer data as it arrives.

Data update options

Here is a summary of how you can update data loaded into BDD, and when each type of update is useful to use.

Options for data updates

To update already loaded data, you have these options:

- Reload data set in Studio
- Refresh data with DP CLI
- Run an incremental update with DP CLI
- When to use each type of update

Updates that you run with DP CLI are also called **scripted updates**.

Reload data set in Studio

The **Reload data set** option in Studio is useful when you want to reload a newer version of the data than you loaded before. It applies to personally uploaded files and to data imported from a JDBC source. Note that this option works only for data sets in Studio's Catalog.

For a diagram of updating data sets that were loaded in Studio, see Studio-loaded files: data update diagram in this guide.

For detailed procedures for loading and reloading data in Studio, see the sections in the *Studio User's Guide*.

Refresh data with DP CLI

The `Refresh data` operation from DP CLI reloads an existing data set in a Studio project, replacing the contents of the data set with the latest data from Hive in its entirety. If the schema in the source Hive table changes, so does the newly-referenced data set. In this type of update, old data is removed and is replaced with new data. New attributes may be added, or attributes may be deleted. Also, the data type for an attribute may change.

For a diagram of updating data sets that were loaded with DP CLI, see DP CLI-loaded files: data update diagram in this guide.

For detailed information on how to run scripted updates with DP CLI, see the *Data Processing Guide*.

Run an incremental update with DP CLI

The `Incremental update` operation from DP CLI lets you add newer data to an existing BDD application, without removing already loaded data. In this type of update, the records' schema cannot change. An incremental update is most useful when you keep already loaded data, but would like to continue adding new data. For example, you can add more recent twitter feeds to the ones that are already loaded.

For a diagram of updating data sets that were loaded with DP CLI, see DP CLI-loaded files: data update diagram in this guide.

For detailed information on how to run scripted updates with DP CLI, see the *Data Processing Guide*.

When to use each type of update

This table summarizes when it is useful to use each type of update.

Type of data update	Useful when...
Reload data set in Catalog (in Studio)	This update is useful when you want to replace the loaded file with an updated version. Similarly, if data in a JDBC source was updated, you can reload it this way.
Scripted updates with DP CLI (`Refresh data` and `Incremental update`)	You can run scripted updates on files that originated from a Studio's upload, and on files that BDD discovers in Hive, when you run its data processing workflow for loading data using DP CLI. You can run either type of the scripted updates periodically, by writing update scripts and `cron` jobs on the Hadoop machines that utilize options for these updates from Data Processing CLI. Depending on characteristics of your source data, you may need to periodically run both types of scripted updates, or only one of them. For example, you may want to create a `cron` job that runs an incremental update nightly. This adds data from that day to the existing data set in a project in Studio. In addition to a periodic incremental update, you can run a `Refresh data` update weekly, to replace the data in the project wholesale with the new data that was collected in Hive during the week. A `Refresh data` update is also useful to run weekly because it lets you handle deletes from the source data set.

7

What's New and Changed in this Release

This section describes the changes made for this release of BDD, including new, deprecated, and unsupported features.

New and updated features

The following features have been added, improved, or updated for this Oracle Big Data Discovery release.

Transform-related features

This release includes these transform-related features and improvements:

- You can switch between row view and value list view in **Transform**. You can toggle the **Transform** data grid view in Studio, to examine attributes in the project data set. The data grid can display either a row view — a list of records from the project data sample, where each row represents a record, or a value list view of attribute values. This view shows attribute value distribution in a column display. It lists unique attribute values found in the data sample, listed in descending order by frequency.
- Improved performance of previews in **Transform**.

Improvements in data scale and management

This release includes these improvements to data and scale management:

- Improved performance of join operations.
- Improved performance of Geotagger lookups. The Geotagger model has been updated to improve its performance. When you update the Geotagger model, you obtain a new database of geonames. For information on how to update the database with geonames, see *The Data Processing Guide*.

Data science capabilities

This release includes these changes that make data science tasks easier for all users in BDD:

- Improved Studio chart component provides a consistent user experience.
- The **Chart** > **Scatter Chart** type in Studio has an additional advanced visualization type, the **Scatterplot Matrix**. The matrix plots three or more metrics against each

other and shows a binned scatterplot and correlation value for each pairing, so that you can visually review the relationships between attributes. For information on using and configuring a **Scatterplot Matrix**, see the *Studio User's Guide*.

8

Where You Go from Here

Quick reference to areas of interest

Use this table to locate information on a specific subject of BDD Cloud Service.

BDD feature	Where to find information on it?
List of known issues	*Known Issues for Oracle Big Data Discovery Cloud Service*
Loading data into BDD with DP CLI	*Data Processing Guide*
Loading data in Studio	*Studio User's Guide*
Updating data sets with DP CLI: `Refresh data` and `Incremental update`	*Data Processing Guide*
Updating data sets in Studio	*Studio User's Guide*
Transforming data	*Studio User's Guide* and *Transform API Reference*
Data processing information in BDD, including default attribute types at loading time, sample size, DP CLI options, Dgraph HDFS Agent, and logs from the Data Processing component.	*Data Processing Guide*
Writing your own Custom Visualization Component for using with Studio	*Extensions Guide*
Administering BDDCS	*Administrator's Guide*

A

Glossary

attribute

An **attribute** consists of a name and values on a record.

Just like columns describe rows in a table, attributes describe records in Big Data Discovery. Each set of attributes is specific to a particular data set of records. For example, a data set consisting of a store's products might contain attributes like "item name", "size", "color", and "SKU" that records can have values for. If you think of a table representation of records, a record is a row and an attribute name is a column header. Attribute values are values in each column.

An attribute's configuration in the schema controls three characteristics of each attribute: required (or not), unique (or not), and having a single or multiple assignments. In other words, an attribute's configuration in the schema determines whether an attribute is:

- Required. For a required attribute, each record must have at least one value assigned to it.
- Unique. For unique attributes, two records can never have the same assigned value.
- Single-value or multi-value (also known as single-assign and multi-assign). Indicates whether a record may have at most one value, or it can have more than one value assignments for the same attribute. Single-value attributes can at most have one assigned value on a record. For example, each item can have only one SKU. Multi-value attributes allow for multiple assigned values on a single record. For example, a Color attribute may allow multiple values for a specific record.

These characteristics of attributes, along with the attribute type, originate in the schema maintained in the Dgraph. In addition, Studio has additional attribute characteristics, such as refinement mode, or a metric flag. Studio also lets you localize attribute descriptions and display names.

Most attributes that appear in Big Data Discovery appear in the underlying source data. Also, in Big Data Discovery you can create new attributes, change, or delete attributes within a project. These changes are not persisted to the source data in Hive. Some attributes are the result of enrichments that Big Data Discovery runs on the data it discovers.

See also data set, Dgraph database, schema, record, type (for attributes), and value.

base view

The base view for a data set represents the fundamental attributes in a project data set. Base views represent the data "as is". You can create custom views, which are useful for data aggregation, computation, and visualization.

Custom views include only data from specific selected attributes (or columns) in the underlying data. They provide different ways of looking at data. Each custom view has a definition expressed as an EQL statement. Custom views do not eliminate the base view, which is always present in the system.

You can create multiple custom views in your project in parallel to each other.

Linked views are created automatically when you join data sets. They are broadened views of data. Linked views widen a base view by joining the original data set with another data set.

BDD Application

A **BDD Application** is a type of BDD project that has special characteristics. An application often contains one or more data sets, where at least one of them may be loaded in full. You can transform and update data in a BDD application. Data updates can be done periodically. You maintain a BDD application for long-lasting data analysis and reporting on data that is kept up-to-date.

As opposed to ad-hoc exploratory BDD projects which any user in BDD can create, BDD administrators own and certify BDD analytic applications, which they can share with their teams.

See also project.

Big Data Discovery Cluster

A **Big Data Discovery Cluster** is a deployment of Big Data Discovery components on any number of nodes.

Nodes in the deployment have different roles:

- **Hadoop nodes** represent machines in the Hadoop cluster. A pre-existing Hadoop cluster is assumed when you deploy Big Data Discovery. Some of the machines from the pre-existing Hadoop cluster may also become the nodes on which components of Big Data Discovery (those that require running in Hadoop) are deployed.

- **WebLogic Server nodes** are machines on which Java-based components of Big Data Discovery — Studio and Dgraph Gateway — run inside WebLogic Server. At deployment time, you can add more than one of these WebLogic Server machines (or nodes) to the BDD cluster.

- **Dgraph-only nodes** represent machines in the Big Data Discovery cluster, which the Dgraph instance is running on. They themselves form a Dgraph cluster within the Big Data Discovery cluster deployment.

You have multiple options for deploying Big Data Discovery in order to use hardware efficiently. For example, you can co-locate various parts of the Big Data Discovery on the same nodes. For information on BDD cluster deployment options, see the *Installation and Deployment Guide*.

Catalog

A Catalog is an area of the Studio application that lists:

- Data sets available to you

- Projects that you have access to

The Catalog contains options to create a new data set, explore a data set, or navigate to an existing project.

When the Data Processing component of Big Data Discovery runs, the available data sets are discovered by Big Data Discovery in the Hive database, profiled, and then presented as a list in the Catalog.

You can then use the Catalog to identify data sets of interest to you, by navigating and filtering data sets and projects based on data set metadata and various characteristics of projects. You can also display additional details about each data set or project, for further exploration.

The first time you log in to Big Data Discovery, the Catalog may display discovered data sets only, but not projects. After you or members of your group create and share projects, the Catalog displays them when you log in, in addition to the available data sets.

See also data set and project.

Custom Visualization Component

A **Custom Visualization Component** is an extension to Studio that lets you create customized visualizations in cases where the default components in Studio do not meet your specific data visualization needs.

custom views

Custom views are useful for data aggregation, computation, and visualization. Compared with base views that contain fundamental data for records, custom views include only data from specific selected attributes (or columns) in the underlying data. This way, custom views provide different ways of looking at data. Each custom view has a definition expressed as an EQL statement.

Custom views do not eliminate the base view, which is always present in the system. You can create multiple custom views in your project in parallel to each other.

See also base view and linked view.

data loading

Data loading is a process for loading data sets into BDD. Data loading can take place within Studio, or with Data Processing CLI.

In Studio, you can load data by uploading a personal file or data from a JDBC source. You can also add a new data set as the last step in transforming an existing data set.

Using DP CLI, you can load data by manually running a data loading workflow, or by adding it to a script that runs on your source data in Hive, and uses whitelists and blacklists, as well as other DP CLI parameters, to discover and load source data into BDD.

Often, you load a sample of data into BDD. You can use an option in DP CLI to change the sample size. Also, in Studio, you can load a full data set into your project created with sampled data. For information on loading full data, see the *Data Exploration and Analysis Guide*.

See also sampling, and data updates.

Data Processing (component of BDD)

Data Processing is a component in Big Data Discovery that runs various data processing workflows.

For example, the data loading workflow performs these tasks:

- Discovery of data in Hive table
- Creation of a data set in BDD

- Running a select set of enrichments on discovered data sets
- Profiling of the data sets
- Indexing (by running the Dgraph process that creates the Dgraph database)

To launch data processing workflows when Big Data Discovery starts, you use the Data Processing Command Line Interface (DP CLI). It lets you launch various data processing workflows and control its behavior. For information, see the *Data Processing Guide*.

See also Dgraph database, enrichments, sampling, and profiling.

data set

In the context of Big Data Discovery, a **data set** is a logical unit of data, which corresponds to source data such as a delimited file, an Excel file, a JDBC data source, or a Hive table.

The data set becomes available in Studio as an entry in the Catalog. A data set may include enriched data and transformations applied to it from **Transform**. Each data set has a corresponding set of files in the Dgraph database.

A data set in Big Data Discovery can be created in different ways:

- When Big Data Discovery starts and you run its data processing workflow for loading data
- When you load personal data files (delimited or Excel files) using Studio
- When you load data from a JDBC data source using Studio
- When you use Transform features to create a new data set after running a transformation script
- When you export a data set from BDD into Hive, and BDD discovers it and adds it to Catalog.

See also sampling, attribute, database, schema, record, type (for attributes), and value.

data set import (personal data upload)

Data set import (or **personal data upload**) is the process of manually creating a data set in Studio by uploading data from an Excel or delimited (CSV) file.

data updates

A **data update** represents a change to the data set loaded into BDD. Several types of updates are supported.

In Studio's Catalog, you can run **Reload data set** for a data set that you loaded from a personal file, or from a JDBC source. This is an update to a personally loaded file or to a sample from JDBC.

Using DP CLI, you can run two types of updates: `Refresh data` and `Incremental update`. These updates are also called scripted updates, because you can use them in your scripts, and run them periodically on data sets in a project in Studio.

The `Refresh data` operation from DP CLI reloads an existing data set in a Studio project, replacing the contents of the data set with the latest data from Hive in its entirety. In this type of update, old data is removed and is replaced with new data. New attributes may be added, or attributes may be deleted. Also, data type for an attribute may change.

The `Incremental update` operation from DP CLI lets you add newer data to an existing BDD application, without removing already loaded data. In this type of update, the records schema cannot change. An incremental update is most useful when you keep already loaded data, but would like to continue adding new data. For example, you can add more recent twitter feeds to the ones that are already loaded.

Dgraph

The **Dgraph** is a component of Big Data Discovery that runs search analytical processing of the data sets. It handles requests users make to data sets. The Dgraph uses data structures and algorithms to provide real-time responses to client requests for analytic processing and data summarization.

The Dgraph stores the database created after source data is loaded into Big Data Discovery. After the database is stored, the Dgraph receives client requests through Studio, queries its database, and returns the results.

The Dgraph is designed to be stateless. This design requires that a complete query is sent to it for each request. The stateless design facilitates the addition of Dgraph processes (during the installation) for load balancing and redundancy — any replica of a Dgraph can reply to queries independently of other replicas.

Dgraph database

The **Dgraph database** represents the contents of a data set that can be queried by Dgraph, in Big Data Discovery. Each data set has its own Dgraph database. The Dgraph database is what empowers analytical processing. It exists both in persistent files in memory and on disk. The database refers to the entire set of files of the data set and to the logical structures into which the information they contain is organized internally. Logical structures describe both the contents and structure of the data set (schema).

The Dgraph database stores data in way that allows the query engine (the Dgraph) to effectively run interactive query workloads, and is designed to allow efficient processing of queries and updates. (The Dgraph database is sometimes referred to as an index).

When you explore data records and their attributes, Big Data Discovery uses the schema and its databases to allow you to filter records, identify their provenance (profiling), and explore the data using available refinements.

See also attribute, data set, schema, record, refinement, type (for attributes), and value.

Dgraph Gateway

The **Dgraph Gateway** is a Java-based interface in Big Data Discovery for the Dgraph, providing:

- Routing of requests to the Dgraph instances
- Caching
- Handling of cluster services for the Dgraph instances, using the ZooKeeper package from Hadoop

In BDD, the Dgraph Gateway and Studio are two Java-based applications co-hosted on the same WebLogic Server.

Discover

Discover is one of the three main modes, or major areas of Studio, along with **Explore** and **Transform**. As a user, you work within one of these three modes at a time.

Discover provides an intuitive visual discovery environment where you can compose and share discovery dashboards using a wide array of interactive data visualization components. It lets you link disparate data sources to find new insights, and publish them across the enterprise using snapshots.

Discover is where you create persistent visualizations of your data and share them with other users of your project.

See also **Explore** and **Transform**.

enrichments

Enrichments are modules in Big Data Discovery that extract semantic information from raw data to enable exploration and analysis. Enrichments are derived from a data set's additional information such as terms, locations, the language used, sentiment, and key phrases. As a result of enrichments, additional derived attributes (columns) are added to the data set, such as geographic data, or a suggestion of the detected language.

For example, BDD includes enrichments for finding administrative boundaries, such as states or counties, from Geocodes and IP addresses. It also has text enrichments that use advanced statistical methods to pull out entities, places, key phrases, sentiment, and other items from long text fields.

Some enrichments let you add additional derived meaning to your data sets. For example, you can derive positive or negative sentiment from the records in your data set. Other enrichments let you address invalid or inconsistent values.

Some enrichments run automatically during the data processing workflow for loading data. This workflow discovers data in Hive tables, and performs data set sampling and initial data profiling. If profiling determines that an attribute is useful for a given enrichment, the enrichment is applied as part of the data loading workflow.

The data sets with applied enrichments appear in Catalog. This provides initial insight into each discovered data set, and lets you decide whether the data set is a useful candidate for further exploration and analysis.

In addition to enrichments that may be applied as part of data loading by Data Processing, you can apply enrichments to a project data set from the **Transformation Editor** in **Transform**. From **Transform**, you can configure parameters for each type of enrichment. In this case, an enrichment is simply another type of available transformation.

See also transformations.

Explore

Explore is an area in Studio where you analyze the attributes and their values for a single data set. You can access **Explore** from the Catalog, or from within a project. You can use **Explore** to analyze attributes and their value distributions for a single data set at a time.

The attributes in **Explore** are initially sorted by name. You can filter displayed attributes, and change the sort order.

For each attribute, **Explore** provides a set of visualizations that are most suitable for that attribute's data type and value distribution. These visualizations let you engage with data to find interesting patterns and triage messy data.

Exploring a data set does not make any changes to that data set, but allows you to assemble a visualization using one or more data set attributes, and save it to a project page.

See also **Discover** and **Transform**.

exporting to HDFS/Hive

Exporting to HDFS/Hive is the process of exporting the results of your analysis from Big Data Discovery into HDFS/Hive.

From the perspective of Big Data Discovery, you are exporting the files from Big Data Discovery into HDFS/Hive. From the perspective of HDFS, you are importing the results of your work from Big Data Discovery into HDFS. In Big Data Discovery, the **Dgraph HDFS Agent** is responsible for exporting to HDFS and importing from it.

The exporting to HDFS process is not to be confused with data set import, also known as personal data upload, where you add a data set to BDD, by uploading a file in Studio (in which case BDD adds a data set to Hive).

linked view

Linked views are created automatically when you join data sets. They are broadened views of data. Linked views widen a base view by joining the original data set with another data set.

See also base view and custom views.

metadata

Each data set includes various types of **metadata** — higher-level information about the data set attributes and values.

Basic metadata is derived from the characteristics of data sets as they are registered in Hive during data processing. This is called **data profiling**. Big Data Discovery performs initial data profiling and adds metadata, such as Geocode values, derived from running various data enrichments.

As you explore and analyze the data in Big Data Discovery, additional metadata is added, such as:

- Which projects use this data set
- Whether the source data has been updated

Some metadata, such as attribute type, or multi-value and single-value for attributes, can be changed in **Transform**. Other metadata uses the values assigned during data processing.

In addition, various types of attribute metadata are available to you in Studio. They include:

- Attribute display names and descriptions
- Formatting preferences for an attribute
- Available and default aggregation functions for an attribute

Oracle Big Data Discovery

Oracle Big Data Discovery is a set of end-to-end visual analytic capabilities that leverage the power of Hadoop to transform raw data into business insight in minutes, without the need to learn complex products or rely only on highly skilled resources.

It lets you find, explore, and analyze data, as well as discover insights, decide, and act.

The Big Data Discovery software package consists of these main components:

- **Studio**, which is the front-end Web application for the product, with a set of unified interfaces for various stages of your data exploration:

  You use the Catalog to find data sets and **Explore** to explore them.

  You can then add data sets to projects, where you can analyze them, or use **Transform** to apply changes to them.

  You can also export data to Hive, for further analysis by other tools such as Oracle R. Both **Explore** and **Transform** are part of the area in the user interface known as **project**. Note that you can explore data sets that are part of a project, as well as source data sets that are not included into any project but appear in **Explore**.

- **Dgraph Gateway**, which performs routing of requests to the Dgraph instances that perform data indexing and query processing.

- The **Dgraph** is the query engine of Big Data Discovery.

- **Data Processing**, which runs various data processing workflows for BDD in Hadoop. For example, for data loading workflow, it performs discovery, sampling, profiling, and enrichments on source data found in Hive.

profiling

Profiling is a step in the data loading workflow run by the Data Processing component.

It discovers the characteristics of source data, such as a Hive table or a CSV file, and the attributes it contains, and creates metadata, such as attribute names, attribute data types, attribute's cardinality (a number of distinct values a record has from the attribute), and time when a data set was created and updated. For example, a specific data set can be recognized as a collection of structured data, social data, or geographic data.

Using **Explore**, you can look deeper into the distribution of attribute values or types.

Using **Transform**, you can adjust or change some of these metadata. For example, you can replace null attribute values with actual values, or fix other inconsistencies, such as change an attribute that profiling judged to be a String value into a number.

project

A BDD **project** is a container for data sets and user-customized pages in Studio. When you work with data sets in BDD, you put them in projects in Studio. In a project, you can create pages with visualizations, such as charts and tables.

As a user in Studio, you can create your own project. It serves as your individual sandbox for exploring your own data. In a project you can try adding different sample data sets, and identify interesting data sets for future in-depth analysis.

BDD projects often, but not always, run on sample data and allow you to load newer versions of sample data into them. Each BDD deployment can support dozens of ad-hoc, exploratory BDD projects for all Studio users. You can turn the most interesting or popular BDD projects into BDD applications.

From within a project, you can:

- Try out an idea on a sample of data

- Explore a data set and answer a simple analytics question

- Transform a data set

- Link data sets
- Create custom views of data set data
- Save and share it with others

See also BDD application.

record

A **record** is a collection of assignments, known as values, on attributes. Records belong to data sets.

For example, for a data set containing products sold in a store, a record can have an item named "T-shirt", be of size "S", have the color "red", and have an SKU "1234". These are **values** on attributes.

If you think of a table representation of records, a record is a row and an attribute name is a column header, where attribute values are values in each column.

record identifier (Studio)

A **record identifier** in Studio is one or more attributes from the data set that uniquely identify records in this data set.

To run an incremental update against a project data set, you must provide a **record identifier** for a data set so that the data processing workflow can determine the incremental changes to update, and you must load the full data set into the project. It is best to choose a record identifier with the highest percentage of key uniqueness (100% is the best).

refinement state

A **refinement state** is a set of filter specifications (attribute value selections, range selections, searches) to narrow a data set to a subset of the records.

sample

A **sample** is an indexed representative subset of a data set that you interact with in Studio. As part of its data loading workflow, Data Processing draws a simple random sample from the underlying Hive table, then creates a database for the Dgraph to allow search, interactive analysis, and exploration of data of unbounded size.

The default size of the sample is one million records. You can change the sample size.

sampling

Sampling is a step in the data loading workflow that Data Processing runs. Working with data at very large scales causes latency and reduces the interactivity of data analysis. To avoid these issues in Big Data Discovery, you can work with a sampled subset of the records from large tables discovered in HDFS. Using sample data as a proxy for the full tables, you can analyze the data as if using the full set.

During its data loading workflow, Data Processing takes a random sample of the data. The default sample size is one million records. You can adjust the sample size. If a source Hive table contains fewer records than the currently specified sample size, then all records are loaded. This is referred to as "data set is fully loaded". Even if you load a sample of records, you can later load a full data set in BDD, using an option in Studio's Data Set Manager.

schema

Schema defines the attributes in the data set, including characteristics of each attribute.

See also attribute, data set, Dgraph database, record, type (for attributes), and value.

scratchpad

The **scratchpad** is a part of **Explore** that lets you quickly compose visualizations using multiple attributes. When you add an attribute to the scratchpad, either by clicking on a tile, or by using typeahead within the scratchpad itself, the scratchpad renders a data visualization based on the attributes in the scratchpad. This lets you concentrate on the data, rather than on configuring this visualization yourself.

In addition to rendering a visualization, the scratchpad provides several alternative visualizations for the attributes, allowing you to quickly switch to an alternative view, without having to change any configuration. From within a project, you can save a scratchpad visualization to a page in **Discover** where you can apply a more fine-grained configuration.

semantic type

A **semantic type** is a setting in Studio that provides additional information about an attribute. It is a logical addition to an attribute that refines how an attribute is used in Studio. You add a semantic type to an attribute and then you can search and navigate based on the semantic type. A semantic type does not change an attribute's data type.

A semantic type can indicate whether an attribute represents an entity (places, people, organizations), personal information (SSN, phone numbers, emails, etc.), units of measure (currency, temperature, etc.), date times (year, month, day, etc.), and digital info (OS versions, IP addresses, etc.) For example, you could add a semantic type of Currency to an attribute named Price and then search and refine the data set by the keyword or value of Currency.

For details about creating semantic types, see the *Studio User's Guide*.

source data

Source data can be a CSV file, an Excel file, a JDBC data source, or a Hive table. All source data is visible in Hadoop, is stored in HDFS, and is registered as a Hive table.

Any source Hive table can be discovered by the data loading workflow that Data Processing (DP) component runs. As part of data loading, DP takes a random sample of a specific size and creates a data set in Dgraph, visible in the Catalog, for possible exploration and selection.

Once a sampled source data appears in the Catalog, it becomes a Big Data Discovery data set, and already represents a sample of the source Hive table.

Here is how BDD interacts with source data sets it finds in Hive:

- BDD does not update or delete source Hive tables. When BDD runs, it only creates new Hive tables to represent BDD data sets. This way, your source Hive tables remain intact if you want to work with them outside of Big Data Discovery.
- Most of the actions in the BDD data set lifecycle take place because you select them. You control which actions you want to run. Indexing in BDD is a step that runs automatically.

Studio

Studio is a component of Big Data Discovery. Studio provides a business-user friendly user interface for a range of operations on data.

Some aspects of what you see in Studio always appear. For example, Studio always includes search, **Explore**, **Transform**, and **Discover** areas. Other parts of your interface you can add as needed. These include many types of data visualization components.

For example, you can add charts, maps, pivot tables, summarization bars, timelines, and other components. You can also create custom visualization components.

Studio provides tools for loading, exploration, updating, and transformation of data sets. It lets you create projects with one or more data sets, and link data sets. You can load more data into existing projects. This increases the corpus of analyzed data from a sample to a fully loaded set. You can also update data sets. You can transform data with simple transforms, and write transformation scripts.

Studio's administrators can control data set access and access to projects. They can set up user roles, and configure other Studio settings.

Projects and settings are stored in a relational database.

token (in Studio)

A **token** is a placeholder (or variable) that Studio uses in the EQL query that powers a custom visualization. It lets you write an abstract EQL query once and provide a way for other users of your Studio project to swap in different values on demand, in place of a token.

Tokens can represent various aspects of an EQL query, such as attributes, views, sorts, or data. For example, using a view token in an EQL query allows project's users to employ the same query multiple times, to visualize different views. In the EQL query syntax in Studio's custom visualizations editor, tokens are strings enclosed by percentage signs (%).

After writing an EQL query, you can request Studio to detect tokens in the EQL script you wrote. You can then designate which of the tokens are intended to represent attributes, views or sorts. Until you designate what each token is for, tokens are unassigned. All tokens except data must be assigned to a query role before the visualization is complete.

See also Custom Visualization Component.

Transform

Transform is one of the three main areas of Studio, along with **Explore** and **Discover**. **Transform** is where you make changes to your project data set. It allows you to edit the data set values and schema, either to clean up the data, or to add additional values.

Transform unlocks data cleansing, manipulation and enrichment activities that are commonly restricted to rigid ETL processes. **Transform** lets you easily construct a transformation script through the use of quick, user-guided transformations as well as a robust, Groovy- based list of custom transform functions.

In **Transform**, you interactively specify a list of default enrichments and transformations, or you can write your own custom transformations. You can preview the results of applying the transformations, then add them to a transformation script that you can edit, run against the project data set, and save.

See also Explore and Discover.

Transformation Editor

The **Transformation Editor** is a part of **Transform** in Studio where you transform your data and often create derived attributes. Along with Groovy support, the **Transformation Editor** gives you access to a list of easy-to-use default transformations (based on Groovy) that let you speed up the process of data conversion, manipulation and enrichment.

Transformation script

A **Transformation script** is a sequential set of transformations organized into a script that you run against a project data set. When you run the transformation script against a project data set, no new entry is created in the Catalog, but the current project does reflect the effects of each transformation step in the script.

After you run a transformation script against project data set, you can also create a new version of the project data set, and publish it to the Catalog. This creates a new full data set in Hadoop, thus unlocking the transformed data for exploration in Big Data Discovery as well as in other applications and tools within Hadoop.

If a transformation script is useful to other Studio users, you can share the script by publishing it, so that it is available to load and run in other projects.

transformations

Transformations (also called transforms) are individual changes to a project data set. For example, you can apply any of the following transformations:

- Change data types
- Change capitalization of values
- Remove records
- Split columns into new ones (by creating new attributes)
- Group or bin values
- Extract information from values

Transformations can be thought of as a substitute for an ETL process of cleaning your data. Transformations can be used to overwrite an existing attribute, modify attributes, or create new attributes.

Most transforms are available directly as unique editors in **Transform**. Some transformations are enrichments.

You can you use the Groovy scripting language and a list of custom, predefined Groovy-based transform functions available in Big Data Discovery, to create a custom transformation.

See also enrichment, Transform, and Transformation Editor.

type (for an attribute)

An attribute's **type** determines the possible values that can be assigned to an attribute. Examples of attribute types include Boolean, Integer, String, Date, Double, and Geocode.

String attributes have additional characteristics related to text search.

See also attribute, data set, Dgraph database, schema, record, and value.

value (for an attribute)

An attribute's **value** is an assignment to an attribute for a specific record.

For example, for a data set containing products sold in a store, a record may have:

- For an attribute with the name "Item Name", an assignment to the attribute's value "t-shirt"
- For an attribute named "Color", an assignment to the attribute's value "red"

- For an attribute named "SKU", an assignment to the attribute's value "1234"

See also attribute, data set, Dgraph database, schema, record, and type (for attributes).

Index