Oracle® Big Data Discovery

Upgrade Guide

Version 1.2.2 • Revision B • October 2016

ORACLE

Copyright and disclaimer

Table of 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 helps you upgrade your Oracle Big Data Discovery cluster and describes the major changes included in the new release.

Audience

This guide is intended for system administrators and developers who are upgrading Oracle Big Data Discovery.

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*.

Chapter 1

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 the Oracle Big Data Discovery 1.2.2 release.

Platform support

This release includes the following platform updates:

- **Updated Hadoop/Spark Support**. Support for BDA 4.5, CDH 5.6 and 5.7.1, and HDP 2.4.x (min. 2.4.2)
- **Support for secure Hadoop environments**:
  - HDFS data at rest encryption
  - TLS/SSL for HDFS, YARN, and Hive

Transform-related features

This release includes the following transform-related features and improvements:

Feature	Description
Streamlined **Transform** menu	The **Transform** page has been redesigned with new attribute menus and a slightly different workflow which is contextual. You select an attribute column and the options available on attribute menu are contextual based on the data type of the attribute you selected. The **Delete Rows** transform has been replaced by the **Filter Rows** transform, and **Aggregate** and **Join** transforms have been added (see the next row in this table for a description). You can select columns from the data table on Transform and see which other transformations you can perform. Also, you can now click **Preview** before running transform operations to see how the data will change if you commit a transform script.
Aggregate and **Join** transforms in **Transform**	The **Aggregate** and **Join** transforms help you run data wrangling at scale, and turn BDD into a powerful tool in your data lab. With aggregations and joins, you can reshape large data sets by running multiple-row transformations in addition to simple transforms and single-row changes, such as trimming. For example, aggregations let you forecast by customer region, or you can join data sets and then explore the newly widened data set in Studio.
Ability to zoom in on segments of full source data sets with **Filter Rows**	You can apply the refinements you have defined by interacting with sample data sets in BDD back to the full source data set using the **Filter Rows** transformations. This provides you with interactive performance at full scale, by letting you apply "pan and zoom" techniques to big data. For example, consider a source data set with 300M weather observation records from across the US. By default, BDD creates a random sample of 1M records from this source data set and presents it as a BDD data set in Studio. By interacting with a map or other Studio's visualization for this data set, you might filter in ("pan") to only records in California in 2016. Next, using **Filter Rows** transform, you could go back to the full source data set to get all of the records in California in 2016 ("zoom"). If the number of records in the "full" source data set that matches the refinements still exceeds the sample size of the data set in BDD (the default is 1M but can be changed), BDD creates a random sample of that subset based on the value for its configured sample size.

Feature	Description
A new **Manage null values** editor added to **Transform**	You can make sure all records in the data set have value assignments for each attribute and add them if they are missing. This is especially useful because in many cases, even though the value is missing because data is messy, you know from context what the value should be. For example, if for a record with an SKU number 123, the attribute Price is empty (because of the messy state of data at loading time), you can fill in the null value for Price with the actual value. The **Manage null values** editor lets you easily achieve uniformity in your sample, and normalize it, so that you can continue to analyze and explore data, or run other transformations on it.

Improvements in data and scale management

This release includes the following changes related to data and scale management:

Feature	Description
Improved Catalog filtering	You can filter projects and data sets in the Studio Catalog using metadata. For example, you can search or filter for projects created by a specific user, or for data sets with a user-specified "My sales data" tag. You can also filter on attribute-level metadata such as whether a data set includes geocode values.
Concurrent data set indexing	In previous releases, the Dgraph created one index (now known as the ***Dgraph database***), for all data sets in BDD. In this release, the following changes occurred: • The Dgraph creates a separate Dgraph database for each data set it stores. • Each Dgraph database may have its own leader Dgraph. This is different from previous releases, where there was only one leader Dgraph in the Dgraph cluster for all data sets. This makes it possible to run indexing in BDD concurrently on multiple data sources, and across multiple data sets that may belong to multiple projects or the same project in Studio.
New `bdd-admin` commands and options	The `bdd-admin` script includes a new command that lets you add Data Processing nodes to your BDD cluster, which previously had to be done manually. The script also includes new options for administering the new Transform Service (internal service in BDD). For more information, see the *Administrator's Guide*.

Feature	Description
A flag to turn off indexing for search in some DP CLI workflows	A new flag, `--disableSearch` in DP CLI turns off Dgraph indexing for search. The data set discovery workflow in Data Processing disables record search and value search on all the attributes, irrespective of the average String length of the values. This flag can be used only for provisioning workflows (for new data sets created from Hive tables) and for refresh workflows (with the `--refreshData` flag). This flag cannot be used in conjunction with the `--incrementalUpdate` flag.
Logs for the FUSE client and Transform Service	BDD logs now include logs for the FUSE client used by the Dgraph and for the Transform Service (which is an internal interface used by BDD for running its operations). These logs can be collected (log harvesting), and rotated.

Data science capabilities

This release includes the following changes that make data science tasks easy for all users in BDD:

Feature	Description
BDD Shell	BDD Shell is an interactive tool designed to work with BDD without the Studio's front-end that exposes BDD concepts, such as views and data sources, and includes a command-line interface (CLI) for managing data sources. Using BDD Shell, you can continue working with data sets obtained from BDD, outside of Studio. BDD shell is installed separately. It is based on Python and supports the Spark Python SDK.
Studio visualization upgrades and Sunburst chart	The interfaces for Studio visualizations have been improved across the board to make them easier to use. This includes expanded support for dragging and dropping attributes between zones and unified configuration. Additionally, the Thematic Map has been moved to the Chart component, and a new Sunburst chart sub-type has been added as a variant of the base Pie chart in order to easily view hierarchical data.
In quick look for **Explore**, variance is replaced with standard deviation	In **Explore** area's quick look, variance is replaced with standard deviation. This improves data filtering in Catalog.
In **Explore** and **Transform**, filtering of data sets and projects is based on attribute metadata	You can now edit project and data set metadata from the Catalog. You can filter the Catalog by selected data sets, attributes, or any metadata that you add. You can change names of attributes and data set, and, as the designated data set curator (a new role in BDD), you can add notes and tags on data sets, and refine on data set tags.

BDD as a complete toolkit in your data lab

This release includes the following changes to turn BDD into a complete toolkit in your data lab:

Feature	Description
Notifications panel in Studio	Studio provides asynchronous updates on the status of long-running transform operations. This lets you continue working with data in Studio while operations run in the background and get notifications of their status and successful completion.
Data curation	You can edit project's and data set's attributes, including the semantic type on attributes. You can also add data curators to projects and data sets, and edit the public copy of any data set in BDD. Also, you can create a copy of an existing data set and that copy preserves the metadata from the original existing data set.
Increased speed of data loading and processing in BDD	BDD now loads data faster: • It loads data sets in parallel. • Loading of a single data set in some cases is up to 20% faster. In addition, Groovy-based transformations now run faster as they utilize system resources more efficiently.
Support for storing Dgraph databases (indexes) in HDFS	The Dgraph databases (indexes) can now be stored on HDFS, instead of a shared NFS (also supported but is now optional). This can be enabled after upgrading. For instructions, see the *Administrator's Guide*. When enabled, the Dgraph runs on Hadoop DataNodes. This gives you more flexibility in deciding how to deploy BDD on multiple servers. For example, some of the HDFS-hosting servers can now run not only parts of BDD, such as Data Processing, but also serve as the shared location for storing the Dgraph databases for your BDD cluster deployment. The location in HDFS where you store the Dgraph databases must be shared between all BDD nodes running the Dgraph.

Terminology changes

This release includes the following terminology changes:

Term in previous release	New term	Description
Dgraph index	***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. 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.
Dgraph index directory	***Dgraph database directory***	The directory where the Dgraph stores its databases. It can be located on shared HDFS or NFS storage.
data set key	***data set logical name***	The data set key is renamed in Studio to be the ***data set logical name***. It is assigned to each data set in Studio and is shown in details for the data set. You need to specify the data set logical name to DP CLI for running data set updates.
N/A	***Semantic types on attributes***	A new term in this release. 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. In this release, you add a semantic type to an attribute, and then 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), units of measure (currency, temperature, etc.), date times (year, month, day, etc), and digital information (OS versions, IP addresses, etc.) For example, you could add a semantic type of Currency to an attribute Price and then search and refine the data set by the keyword or value of Currency.

Documentation changes

This release includes the following documentation changes:

Feature	Description
New guides	The BDD documentation set includes two new guides: • The *Security Guide* provides a summary of all security-related features in BDD. • The *BDD Shell Guide* describes how to use the BDD Shell utility.
Renamed guides	The following guides have been renamed: • The *Data Exploration and Analysis Guide* is now the *Studio User's Guide*. • The *Installation and Deployment Guide* is now the *Installation Guide*. • The *Licensing Information User Manual* is the new name of the guide that includes licensing information for Oracle Big Data Discovery.

Unsupported and deprecated features

The following features are not supported in this release, or have been deprecated.

Feature that is not supported	Description
Dgraph Gateway Command Utility	The Dgraph Gateway Command Utility has been removed from BDD. This utility, also called the `endeca-cmd` utility, included these commands: `allocate-bulk-load-port`, `dump-session`, `list-compute-nodes`, `version`, and `warm-cache`.
BDD Studio Server Administration	The **Data Migration**, **File Uploads**, **Mail**, **OpenOffice**, and **Shutdown** tabs have been removed from the **Server Administration** page. **Email Settings** are now available under **Platform Settings**.
Platform support for older versions of software	Support for earlier versions of CDH, HDP, and Spark has been dropped. See the *Installation Guide* for a complete list of supported platforms.
Strict Dgraph requirement for shared NFS storage	The Dgraph no longer strictly requires a shared NFS to store its databases (indexes) on. These can now be stored on HDFS. For more information, see the *Installation Guide* and the *Administrator's Guide*.
Enterprise Manager Cloud Control	In this release, BDD does not support Enterprise Manager. You must use the `bdd-admin` script to administer your BDD cluster.

Chapter 2

Before You Upgrade

Before upgrading BDD, you must download the required software packages and prepare your cluster.

Supported upgrade paths

The following upgrade paths are supported.

- 1.0 to 1.2.2
- 1.1.x to 1.2.2
- 1.2.x to 1.2.2

Upgrade requirements

Before upgrading, make sure your system meets the following requirements.

Requirement	Description
Hadoop	BDD 1.2.2 supports the following: • Cloudera Distribution for Hadoop (CDH) 5.5.x (min. 5.5.2), 5.6, 5.7.1 • Hortonworks Data Platform (HDP) 2.3.4.17-5, 2.4.x (min. 2.4.2) If you don't have one of the above installed, upgrade your Hadoop cluster before upgrading BDD. For instructions, refer to the documentation for your Hadoop distribution. Before upgrading your Hadoop cluster, be aware of the following: • You can't switch to a different Hadoop distribution without reinstalling BDD. For example, if you currently have CDH, you can't switch to HDP. • You should stop your BDD cluster before you upgrade Hadoop. Once Hadoop has been upgraded, follow the procedure described in "Switching Hadoop versions" in the *Administrator's Guide* to enable BDD to work with the new version. • After you upgrade Hadoop, verify that the YARN configuration changes you made before installing BDD weren't reset. For more information, see "YARN setting changes" in the *Installation Guide*. If you're upgrading from BDD 1.0, upgrade Hadoop just before you run the BDD upgrade script. BDD 1.0 doesn't support any of the currently supported versions of CDH, so your BDD cluster must remain stopped once you upgrade Hadoop. Additionally, you should verify that the server roles in your Hadoop cluster don't change during the upgrade.
Secure Hadoop	BDD 1.2.2 can run on Hadoop clusters secured with TLS/SSL and HDFS data at rest encryption. You can configure these in your Hadoop cluster before upgrading BDD, and then enable them for BDD at upgrade time. See the section "Security options" in the *Installation Guide* for instructions. Also, be sure to set the `HADOOP_CERTIFICATION_PATH` property in `bdd.conf` before upgrading.
Operating system	BDD supports the following operating systems: • Oracle Enterprise Linux 6.7+, 7.1 • Red Hat Enterprise Linux 6.7+, 7.1 If you don't have one of the above installed, upgrade your OS before upgrading BDD. Additionally, you must clear the `$http_proxy` environment variable: `export http_proxy=`

Requirement	Description
Dgraph database (index)	The Dgraph database (index) directory must contain enough free space to double your databases, as this may occur temporarily during the upgrade process.  **Note:** Although BDD 1.2.2 enables you to store your Dgraph databases (index) on HDFS, you can't move them there until after the upgrade. For instructions on doing this, see the *Administrator's Guide*.
Transform Service	If you're upgrading from 1.0 or 1.1.x, the upgrade script will install the Transform Service on the nodes you specify in `bdd.conf`. For best performance, these should be WebLogic Managed Servers. The Transform Service requires at least 11GB of RAM and 39GB of virtual memory, and may require more depending on the size of its workloads. Be sure that the nodes you install it on meet these requirements.
Perl modules	Verify that the following Perl modules are installed on the Admin Server: • `Mail::Address` • `XML::Parser` • `JSON-2.90` If any are missing, install them according to the instructions in "Installing the required Perl modules" in the *Installation Guide*.
Other requirements	curl 7.19.7+ and Network Security Services (NSS) 3.16.1+ must be installed on all Studio nodes. Additionally, curl must support the option `--tlsv1.2`.

Downloading the upgrade packages

Once you have satisfied all prerequisites, you can download the upgrade packages.

These are available on the Oracle Software Delivery Cloud. After downloading them, you must put them in a single directory on the Admin Server, called the *upgrade source directory*. You will perform the entire upgrade process from this location.

To download the upgrade packages:

1. On the Admin Server, create a new directory or select an existing one to be the upgrade source directory.
2. Within the upgrade source directory, create a subdirectory named `packages`.
3. Go to the *Oracle Software Delivery Cloud* and sign in.
4. Accept the Export Restrictions.
5. Check **Programs** if it isn't already selected.
6. In the **Product** text box, enter `Oracle Big Data Discovery`.
7. Click **Select Platform** and check **Linux x86-64**.

   Oracle Big Data Discovery displays in the **Selected Products** table.

8. Click **Continue**.
9. Verify that **Available Release** and **Oracle Big Data Discovery 1.2.x.x.x for Linux x86-64** are both checked, then click **Continue**.
10. Accept the Oracle Standard Terms and Restrictions and click **Continue**.
11. In the **File Download** popup, click **Download All**.

    The following packages are downloaded to your machine:
    - **First of two parts of the Oracle Big Data Discovery binary**
    - **Second of two parts of the Oracle Big Data Discovery binary**
    - **Installer for Oracle Big Data Discovery**
    - **SDK for Oracle Big Data Discovery**
    - **Documentation for Oracle Big Data Discovery**

    Also, make a note of each package's part number, as you will need this information to identify it.
12. On the Admin Server, move the packages you downloaded to `/<upgrade_source_dir>/packages`.
13. Rename the first BDD binary package `bdd1.zip` and the second `bdd2.zip`.

    This is required for the upgrade script to recognize them.
14. Move up a directory to the upgrade source directory and unzip the installer package:

    ```
    unzip packages/<installer_package>.zip
    ```

    This creates a new directory within the upgrade source directory called `/installer`, which contains the scripts and files required to perform the upgrade.

Next, you can optionally download a WebLogic Server patch for the upgrade script to apply. If you don't want to patch WebLogic Server, move on to *Obtaining the Hadoop client libraries on page 17*.

Downloading a WebLogic Server patch

You can optionally download a WebLogic Server patch for the upgrade script to apply when it runs.

You can only apply one patch when upgrading. If the patch fails, the upgrade script will remove it and continue running. For more information on patching WebLogic Server, see *Oracle Fusion Middleware Patching with OPatch*.

To download a WebLogic Server patch:

1. Within the upgrade source directory, create a new directory called `WLSPatches`.

   Don't change the name of this directory or the upgrade script won't recognize it.
2. Go to *My Oracle Support* and log in.
3. On the **Patches & Updates** tab, find and download the patch you want to apply.
4. Move all ZIP files associated with the patch to `<upgrade_source_dir>/WLSPatches`.

   Don't extract the files. The upgrade script will do this when it runs.

Next, you must obtain the Hadoop client libraries.

Obtaining the Hadoop client libraries

Next, obtain the Hadoop client libraries and put them on the Admin Server.

BDD requires a number of client libraries to interact with Hadoop. In a normal Hadoop cluster, these libraries are spread out, making it difficult for BDD to find them all. To solve this issue, the upgrade script adds the required libraries to a single JAR, called the Hadoop fat JAR, and distributes it to all BDD nodes.

The specific libraries you need depend on your Hadoop distribution. The location you put them in is arbitrary, as you will define it in `bdd.conf`.

Note: If you're upgrading from BDD 1.0, be sure to obtain the libraries for one of the currently supported CDH versions, even though you haven't upgraded to it yet.

- **CDH:** Download the following files from *http://archive-primary.cloudera.com/cdh5/cdh/5/* to the Admin Server and extract them:
  - `spark-<spark_version>.cdh.<cdh_version>.tar.gz`
  - `hive-<hive_version>.cdh.<cdh_version>.tar.gz`
  - `hadoop-<hadoop_version>.cdh.<cdh_version>.tar.gz`
  - `avro-<avro_version>.cdh.<cdh_version>.tar.gz`

  Be sure to download the files that correspond to the component versions you currently have installed (unless you're upgrading from BDD 1.0).
- **HDP:** Copy the following libraries from your Hadoop nodes to the Admin Server. Note that these directories might not all be on the same node.
  - `/usr/hdp/<version>/hive/lib/`
  - `/usr/hdp/<version>/spark/lib/`
  - `/usr/hdp/<version>/hadoop/`
  - `/usr/hdp/<version>/hadoop/lib/`
  - `/usr/hdp/<version>/hadoop-hdfs/`
  - `/usr/hdp/<version>/hadoop-hdfs/lib/`
  - `/usr/hdp/<version>/hadoop-yarn/`
  - `/usr/hdp/<version>/hadoop-yarn/lib/`
  - `/usr/hdp/<version>/hadoop-mapreduce/`
  - `/usr/hdp/<version>/hadoop-mapreduce/lib/`

If you're upgrading from 1.0 or 1.1.x, you should now apply the upgrade hotfix. If you have 1.2.x, move on to *Backing up your current cluster on page 18*.

Running the BDD hotfix

If you're upgrading from 1.0, 1.1.0 or 1.1.1, you must apply one of the hotfixes to install the backup and restore scripts required during the upgrade.

Note: This isn't required if you're upgrading from 1.1.3 or higher.

There are separate hotfixes for versions 1.0, 1.1.0, and 1.1.1. Run the script that corresponds to the version of BDD you currently have installed.

To run the BDD hotfix:

1. On the Admin Server, go to `<upgrade_source_dir>/installer/hotfix/<version>/hotfix_EADMIN-1503`.

   Where `<version>` is the version of BDD you currently have installed.

2. Run the hotfix script:

   ```
   ./hotfix_EADMIN-1503.sh <path/to/bdd.conf>
   ```

   Where `<path/to/bdd.conf>` is the absolute path to your current `bdd.conf` file.

The hotfix adds two new scripts to `$BDD_HOME/BDD_manager/bin`: `bdd-backup.sh` and `bdd-restore.sh`.

Backing up your current cluster

Next, back up your current cluster.

If you're upgrading from 1.0, 1.1.0, or 1.1.1, run the backup script added by the hotfix. If you're upgrading from 1.1.3 or higher, use `bdd-admin`'s `backup` command. All scripts back up the following data to a single TAR file:

- Configuration files
- Studio database
- Schema and data for Hive tables created in Studio
- Dgraph databases (index)
- Sample files in HDFS

You can use this file to restore your current cluster if the upgrade fails.

Before you run the backup script, verify the following:

- The `BDD_STUDIO_JDBC_USERNAME` and `BDD_STUDIO_JDBC_PASSWORD` environment variables are set. If they aren't, the script will prompt you for the username and password of the Studio database at runtime.
- The database client is installed on the Admin Server. For MySQL databases, this should be MySQL client. For Oracle databases, this should be Oracle Database Client, which must be installed with a type of Administrator. Note that the Instant Client isn't supported.
- If you have an Oracle database, the `ORACLE_HOME` environment variable is set to the parent directory of the `/bin` directory the `sqlplus` executable is located in. For example, if the `sqlplus` executable is located in `/u01/app/oracle/product/11/2/0/dbhome/bin`, set `ORACLE_HOME` to `/u01/app/oracle/product/11/2/0/dbhome`.

To back up your current cluster:

1. On the Admin Server, open a command prompt and go to `$BDD_HOME/BDD_manager/bin`.
2. Stop your cluster :

   ```
   ./bdd-admin.sh stop [-t <minutes>]
   ```

3. Run the backup script.
   - BDD 1.0:
     ```
     ./bdd-backup.sh -v <backup_tar_file>
     ```
   - BDD 1.1.0 and 1.1.1:
     ```
     ./bdd-backup.sh -o -v <backup_tar_file>
     ```
   - BDD 1.2.x:
     ```
     ./bdd-admin.sh backup -o -v <backup_tar_file>
     ```

   Where `<backup_tar_file>` is the absolute path to the TAR file your cluster will be backed up to. This file must not exist and its parent directory must be writable.
4. Enter the username and password for the Studio database, if prompted.

The script backs up your current cluster to the specified TAR file.

Chapter 3

Upgrading Big Data Discovery

Once you've obtained the required BDD and Hadoop packages and prepared your cluster, you can begin upgrading BDD.

Overview

You upgrade your cluster by running two separate scripts.

The first script merges the current version of `bdd.conf` with the version from the new release. This ensures that your cluster will retain most of its current configuration. However, you still need to manually edit the file to set new properties added in this release and any that couldn't be merged.

The second script upgrades your cluster and installs the new components and features. When it finishes running, your cluster will be completely upgraded and ready to use.

$BDD_HOME

The upgrade script installs the new version of BDD in `/BDD_<version>`, which is the new `$BDD_HOME`. The script doesn't remove `/BDD_<old_version>`, however. You can delete this if you want.

Important: If you decide to uninstall BDD, the uninstallation script will delete the contents of all instances of `/BDD_<version>`, not just the latest one. If there's anything in an older `/BDD_<version>` that you want to keep, back it up to a different location before uninstalling. See the *Installation Guide* for more information.

Silent upgrade

You can optionally run the upgrade script in silent mode. This means that instead of prompting you for information it requires at runtime, the script obtains it from environment variables you set beforehand.

Normally, when you run the script, it prompts you to enter:

- The username and password for your Hadoop cluster manager (Cloudera Manager or Ambari).
- The username and password for the WebLogic Server admin.
- The username and password for the Studio database.

You can avoid these steps by setting the following environment variables before running the script.

Environment variable	Description
`BDD_HADOOP_UI_USERNAME`	The username for Cloudera Manager/Ambari.
`BDD_HADOOP_UI_PASSWORD`	The password for Cloudera Manager/Ambari.
`BDD_WLS_USERNAME`	The username for the WebLogic Server admin.
`BDD_WLS_PASSWORD`	The password for the WebLogic Server admin.
`BDD_STUDIO_JDBC_USERNAME`	The username for the Studio database.
`BDD_STUDIO_JDBC_PASSWORD`	The password for the Studio database.

Merging the configuration file

The first step in the upgrade process is to merge your current `bdd.conf` with the version from the new release.

You do this by running `merge-bddconf.sh` which, populates the new version of `bdd.conf` with values from the current version. Although the merged file will contain most of your current settings, you'll need to manually edit any properties the script couldn't merge, as well as those that were added in the new release.

To merge the configuration files:

1. On the Admin Server, open a command prompt and go to the upgrade source directory.
2. Run the merge script:

   ```
   ./merge-bddconf.sh <current_bdd.conf> <new_bdd.conf>
   ```

   Where `<current_bdd.conf>` is the absolute path to your current `bdd.conf`, and `<new_bdd.conf>` is the relative path to the new one. On most systems, this should be:

   ```
   ./merge-bddconf.sh $BDD_HOME/BDD_manager/conf/bdd.conf bdd.conf
   ```

Next, update the new and unmerged properties in `bdd.conf`.

Editing the configuration file

After you merge `bdd.conf`, you need to manually edit the new properties and any that weren't merged properly.

The upgrade script validates `bdd.conf` at runtime and fails if the file contains any invalid values. To avoid this, keep the following in mind when editing the file:

- The accepted values for some properties are case-sensitive and must be entered exactly as they appear in the table below.
- All hostnames must be fully qualified domain names (FQDNs).
- Any symlinks included in paths must be identical on all nodes. If any are different or don't exist, the upgrade may fail.
- Each port setting requires a unique value. You can't use the same port number more than once.
- Some of the directories defined in `bdd.conf` have location requirements. These are specified in the table below.

The properties you need to edit are described below. You should also review the rest of the file to verify that all other settings are still accurate. Additional information on the properties in `bdd.conf` is available in the *Installation Guide*.

Property	Description
`BDD_OLD_CONFIG`	The absolute path to your current version of `bdd.conf`. This will already be populated.
`INSTALLER_PATH`	The absolute path to the BDD software packages in the upgrade source directory.
`HADOOP_CLIENT_LIB_PATHS`	A comma-separated list of the absolute paths to the Hadoop client libraries. To set this property, copy the template to `HADOOP_CLIENT_LIB_PATHS` and update the paths to point to the client libraries you copied to the install machine. Don't change the order of the paths in the list as they *must* be specified as they appear. For more information on the client libraries and how to obtain them, see *Obtaining the Hadoop client libraries on page 17*.
`HADDOP_CERTIFICATION_PATH`	Only required for Hadoop clusters with TLS/SSL enabled. The absolute path to the directory on the install machine where you put the certificates for HDFS, YARN, Hive, and the KMS services. For instructions on exporting these, see the *Installation Guide*.
`STUDIO_JDBC_URL`	The JDBC URL for your database. This isn't a new property, but the URL for your database may have changed since you installed. You should verify that this property is still accurate and edit it if it's not.

Property	Description
ZOOKEEPER_INDEX	The index of the Dgraph cluster in the ZooKeeper ensemble, which ZooKeeper uses to identify it. This used to be called COORDINATOR_INDEX.
SPARK_ON_YARN_JAR	The absolute path to the Spark on YARN JAR on Hadoop nodes. Verify that the value matches the correct version of your Hadoop distribution.
TRANSFORM_SERVICE_SERVERS	A comma-separated list of the Transform Service nodes. For best performance, these should all be Managed Servers. In particular, they shouldn't be Dgraph nodes, as both the Dgraph and the Transform Service require a lot of memory. Each of these nodes must have at least 11GB of RAM and 39GB of available virtual memory.
TRANSFORM_SERVICE_PORT	The port the Transform Service listens on for requests from Studio.
FORCE	Determines whether the upgrade script will remove files and directories left over from previous installations when it runs. This can be set to TRUE or FALSE.
DGRAPH_USE_MOUNT_HDFS	Specifies whether the Dgraph databases are stored on HDFS. When upgrading from 1.1.x and lower, this must be set to FALSE. When upgrading from 1.2.x, this value can't be changed.
DGRAPH_HDFS_MOUNT_DIR	The absolute path to the local directory where the Dgraph mounts the HDFS root directory. Use a nonexistent directory when installing. If this location changes after installing, the new location must be empty and have read, write, and execute permissions for the bdd user. This setting is only required if DGRAPH_USE_MOUNT_HDFS is set to TRUE.
DGRAPH_ENABLE_CGROUP	Enables cgroups for the Dgraph. This must be set to TRUE if you use cgroups on your Dgraph nodes. If set to TRUE, DGRAPH_CGROUP_NAME must also be set. For more information on setting up cgroups for the Dgraph, see the *Installation Guide* and the *Administrator's Guide*.
DGRAPH_CGROUP_NAME	The name of the cgroup that controls the Dgraph. This is required if DGRAPH_ENABLE_GROUP is set to TRUE.
BACKUP_LOCAL_TEMP_FOLDER_PATH	The absolute path to the default temporary folder on the Admin Server used during backup and restore operations. This can be overridden on a case-by-case basis by the bdd-admin script.

Property	Description
BACKUP_HDFS_TEMP_FOLDER_PATH	The absolute path to the default temporary folder on HDFS used during backup and restore operations. This can be overridden on a case-by-case basis by the bdd-admin script.

BDD 1.0 upgrades

If you're upgrading from BDD 1.0, you should also verify the following properties.

Property	Description
YARN_QUEUE	The YARN queue that Data Processing jobs are submitted to.
DP_ADDITIONAL_JARS	Optional. A comma-separated list of the absolute paths to custom SerDe jars you want to use during data processing. These will be added to the CLI classpath. Note that you must manually copy each SerDe jar to the same location on all cluster nodes before upgrading.

Upgrading the cluster

Once you've updated bdd.conf, you can upgrade your cluster.

Note: If you're upgrading from 1.0, you should upgrade your Hadoop cluster before running the upgrade script.

Before you run the upgrade script, verify that:

- You downloaded the required BDD upgrade packages.
- You moved the Hadoop client libraries to the Admin Server.
- You applied the hotfix (if necessary).
- You backed up your cluster.
- You merged and edited bdd.conf.
- Perl is installed on the Admin Server.
- Your Hadoop cluster is running.

To upgrade your cluster:

1. On the Admin Server, open a command prompt and go to the upgrade source directory.
2. Run the upgrade script:

   ```
   ./upgrade.sh bdd.conf
   ```

3. If you're not running the script in silent mode, enter the following when prompted:
   - The username and password for your Hadoop cluster manager (Cloudera Manager or Ambari).

- The username and password for the WebLogic Server admin.
- The username and password for the Studio database.

Troubleshooting a failed upgrade

If the upgrade fails, you should first determine why it failed. To do this, check the script's log files in `<upgrade_source_dir>/packages/installer/upgrade.log.xxx`.

Studio fails to start

Studio fails to start

In some cases, Studio may fail to start after the upgrade.

If this occurs, restart Studio by going to `$BDD_HOME/BDD-manager/bin` on the Admin Server and running:

```
./bdd-admin.sh restart -c bddServer
```

Rolling back a failed upgrade

Once you've determined why the upgrade failed, you can roll it back and restore your old cluster.

The rollback script removes all of the changes made by the upgrade script. It also restores your old cluster from backup, starts it, and performs a healthcheck.

Important: The rollback script removes the logs created by the upgrade script, so be sure to check them beforehand to determine why the upgrade failed.

When the script finishes running, your cluster will be back to its pre-upgrade state and running. You can then either rerun the upgrade process or restore your old cluster.

If you decide to rerun the upgrade process, you must rerun the merge script, as the merged version of `bdd.conf` wasn't included in the backup.

To roll back a failed upgrade:

1. If you upgraded from BDD 1.0, downgrade your Hadoop cluster to the version you were running before.

   This is required because BDD 1.0 doesn't support CDH 5.5.2. For instructions, refer to the documentation for your Hadoop distribution.
2. On the Admin Server, open a command prompt and go to the upgrade source directory.
3. Run the rollback script:

   ```
   ./rollback.sh bdd.conf [--silent]
   ```

   You can optionally include the `--silent` flag to avoid the confirmation step.
4. Confirm that you want to roll back the upgrade.
5. When prompted, enter the absolute path to your old cluster's backup TAR file.

Chapter 4

After You Upgrade

This section describes some of the changes made to your cluster, as well as tasks you should perform immediately after upgrading.

Verifying the upgrade

After the upgrade script completes successfully, you should verify your upgrade by running the `health-check` script.

To verify the upgrade:

1. On the Admin Server, open a new terminal window and go to `$BDD_HOME/BDD_manager/bin`.
2. Run the `health-check` script:

   ```
   ./bdd-admin.sh status --health-check
   ```

   If your BDD cluster is healthy, the script's output should be similar to the following:

   ```
   [2016/03/24 04:18:55 -0700] [Admin Server] Checking health of BDD cluster...
   [2016/03/24 04:20:39 -0700] [web009.us.example.com] Check BDD functionality......Pass!
   [2016/03
   /24 04:20:39 -0700] [web009.us.example.com] Check Hive Data Detector health......Hive Data
   Detector has previously run
   [2016/03/24 04:20:39 -0700] [Admin Server] Successfully checked statuses.
   ```

Changes made by the upgrade script

The upgrade script made a number of changes to your cluster, but there are a few in particular that you should be aware of.

- Your Dgraph database (index) was upgraded to the new format and `/DGRAPH_INDEX_DIR` now includes a separate database directory for each project. For more information, see the *Administrator's Guide*.
- Your sample files were upgraded to the new format and moved from `$edpDataDir/.collectionData/<collectionName>` to `$edpDataDir/.collectionData/<databaseName>.<collectionName>`. For more information, see the *Data Processing Guide*.

- The list of supported Dgraph flags has changed in this release—some flags were renamed and a few were removed. Any that you had added to the `DGRAPH_ADDITIONAL_ARG` property in `bdd.conf` were updated accordingly. For more information, see the *Administrator's Guide*.

Post-upgrade configuration

If you run into any performance issues after you upgrade, you might need to adjust the configuration of your BDD or Hadoop cluster.

Please refer to the *Installation Guide* for more information on the changes you should make.

Clearing browser cache

After an upgrade, all BDD users should clear their browser caches before logging in to Studio. This ensures they'll be able to open their projects successfully.

Rewriting custom transformations from BDD 1.0

Many of BDD's transformation functions have been refactored since 1.0 was released; however, the upgrade script doesn't update your custom transformation scripts accordingly. Because of this, if you upgraded from BDD 1.0, you need to update your scripts manually before you can use them in your projects.

You can edit your transformation scripts in Studio's **Transform** component. For more information, see the *Studio User's Guide*.

The following table lists the changes made to the functions.

Note: You only need to make these changes if you upgraded from BDD 1.0.

Function in release 1.0.0	Changes

`geotagIPAddressGetCity`
`geotagIPAddressGetCountry`
`geotagIPAddressGetPostCode`
`geotagIPAddressGetRegion`
`geotagIPAddressGetRegionID`
`geotagIPAddressGetSubRegion`
`geotagIPAddressGetSubRegionID`
	Replaced by the `geotagIPAddress` function.

Function in release 1.0.0	Changes

`geotagAddressGetCity`
`geotagAddressGetCountry`
`geotagAddressGetPostcode`
`geotagAddressGetRegion`
`geotagAddressGetSubRegion`
`geotagAddressGetRegionID`
`geotagAddressGetSubRegionID`
	Replaced by the `geotagUnstructuredAddress` function.

`getLocationEntities`
`getOrganizationEntities`
`getPersonEntities`
	Replaced by the `getEntities` function.