Oracle® Big Data Discovery

Security Guide

Version 1.3.2 • August 2016

Copyright and disclaimer

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

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 describes the security features provided and supported by Oracle Big Data Discovery.

Audience

This guide is intended for users responsible for system security, including system administrators, Big Data Discovery administrators, and users who create and configure Big Data Discovery projects.

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.

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.

Path variable	Meaning
`$BDD_HOME`	Indicates the absolute path to your Oracle Big Data Discovery home directory, `$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

Introduction

The security of your Oracle Big Data Discovery (BDD) installation is highly important. Lapses in security can leave your system, your data, and your users vulnerable to exploitation and attack. Fortunately, there are a number of ways you can keep your installation safe using BDD's own security features as well as third party products.

This guide describes the supported methods for keeping your BDD installation and data safe, including:

- **Securing BDD through your operating system.** There are many ways to keep BDD safe at the operating system (OS) level, like using file permissions to control access to BDD's binaries and administration scripts. For more information, see *Operating System-Level Security on page 6*.
- **Maintaining network security.** Firewalls, reverse proxy servers, and Kerberos can all be used to protect BDD communications. Additionally, WebLogic Server and Jetty, which host some of the BDD components, offer security options of their own. For more information, see *Network Security on page 8*.
- **Managing access to your data.** BDD includes features for controlling what data is available to the application and which users have access to it. For more information, see *Data Set Security on page 11*.
- **Controlling access to Studio.** Studio's built-in user management tools let you set access permissions for individual users and groups of users. You can also configure user authentication within Studio itself, or through an existing LDAP or SSO system. For more information, see *Studio User Access on page 16*.

These options can be used in conjunction with your site's own security policies to keep your system safe. Additionally, if you're responsible for security at your site, you should register your BDD installation with *My Oracle Support* and visit the *Critical Patch Updates and Security Alerts page*.

Chapter 2

Operating System-Level Security

This section describes how to protect your BDD installation at the operating system (OS) level.

OS user accounts

You should limit the number of OS users on BDD nodes to minimize the risk of an unauthorized person gaining access to them.

The following table describes the users BDD requires. You should avoid creating more than these, if possible.

Account	Description
Administrator	Each BDD node should have at least one user with administrator privileges. Oracle recommends having two to provide redundancy. For additional security, avoid choosing obvious names such as `system`, `admin`, or `administrator` for your administrator user accounts.
`bdd` user	A dedicated BDD user is required on the node the installation process is run from (later called the WebLogic Admin Server). This user is referred to as the `bdd` user, and can be either an existing user or a new one. In addition to its other requirements, it must have passwordless `sudo` enabled on all BDD nodes. The `bdd` user performs the installation. After installing, it also runs all BDD processes and typically becomes the owner of the DP CLI and the BDD Shell. Because this user is so powerful, you should treat it as a special account and limit its use to as few people as possible. More information on the `bdd` user is available in the *Installation Guide*.
BDD Shell group members	Optional. If you install the BDD Shell application, Oracle recommends you create a dedicated BDD Shell group with limited permissions. Only members of this group are allowed to use the BDD Shell. For more information, see *File permissions on page 8* and the *BDD Shell Guide*.

File permissions

You should use OS file permissions to restrict user access to BDD files. You can control permissions with `chmod`, `umask`, or a similar utility.

The `bdd` user requires read, write, and occasionally execute permissions for all BDD files and directories, as well as the Dgraph databases. You can give other users access to these as well, but you should make sure they can be trusted and be careful about the specific permissions you grant them.

Additionally, you should take special care with the following BDD utilities.

bdd-admin permissions

The `bdd-admin` script is used to perform administrative tasks, such as starting and stopping BDD components and updating your cluster's configuration. It's located in `$BDD_HOME/bdd_admin/bin` on all BDD nodes. However, it can only be run from the Admin Server and therefore must be run by a user that has passwordless `sudo` enabled on all nodes in the BDD cluster.

By default, `bdd-admin` can only be run by the `bdd` user. Oracle strongly advises against enabling other users to run it, as doing so would greatly increase the risk of an intruder gaining access to your data.

More information about the `bdd-admin` script is available in the *Administrator's Guide*.

Data Processing CLI permissions

The DP CLI utility is used to launch Data Processing workflows, either manually or via cron job. By default, it can only be run by the `bdd` user; however, you can give other users permission to run it as well. Be aware, though, that doing so would give those users greater access to your data as well as control over the specific data available to BDD. You should therefore be cautious about which users you grant access to it.

You should also restrict access to the DP CLI's whitelist and blacklist, which are located in `$BDD_HOME/dataprocessing/edp_cli/config/`. These files control which Hive tables the DP CLI processes. For more information, see *Data set whitelists and blacklists on page 13*.

More information on the DP CLI and its whitelist and blacklist is available in the *Data Processing Guide*.

BDD Shell permissions

The BDD Shell is a programming shell used to explore the internals of BDD, interact with Hadoop, and analyze data. It's an optional component that must be installed separately from the rest of BDD.

The BDD Shell can only be used by the `bdd` user and members of a specific OS group that you define when you install it. You can later add other users to the BDD Shell group, but you should ensure that they are trusted and be careful about which permissions you grant them.

More information about the BDD Shell can be found in the *BDD Shell Guide*.

Chapter 3

Network Security

This section describes ways you can keep your network secure.

Firewalls

Oracle recommends using a firewall to protect your network from external entities.

A firewall limits traffic into and out of your network, creating a secure barrier around it. It can consist of a combination of software and hardware, including routers and dedicated gateway machines.

There are multiple types of firewalls, so you should choose one suited to your resources and specific needs. Oracle recommends using a reverse proxy server as part of your firewall. For more information, see *Reverse proxy servers on page 9*.

For more information on using firewalls with BDD, see the *Installation Guide*.

Reverse proxy servers

Using a reverse proxy server in front of Studio adds another layer of protection to BDD by preventing users from accessing your servers directly.

A reverse proxy server sits in front of your other servers (the proxied servers) and assumes their public hostname. Clients can then access content on the proxied servers through the reverse proxy server without ever knowing the servers' real hostnames, which are often internal and sensitive.

Reverse proxy servers are commonly implemented to provide:

- Security or firewalling
- SSL termination
- Load balancing and failover
- Resource caching/acceleration
- URL partitioning

In the context of BDD, a reverse proxy server enables end users to access Studio without directly connecting to the machines running it. For instructions on configuring this, see the *Installation Guide*.

TLS/SSL

The TLS (Transport Layer Security) and SSL (Secure Sockets Layer) protocols provide end-to-end encryption for communications between applications over a network. (Note that TLS is technically the replacement of SSL, but both are commonly referred to as SSL.)

Although BDD doesn't currently support TLS/SSL for its communications with Hadoop or for the communications between its components, you can enable it on Studio's outward-facing ports to encrypt user traffic. This can be done in one of the following ways:

- Through WebLogic Server. You can do this through BDD's configuration file. This method activates WebLogic's default demo keystores, which you should replace with your own certificates after deployment.
- Through a reverse proxy server.

For more information on BDD's TLS/SSL options, see the *Installation Guide*.

TLS/SSL in Hadoop

BDD can run on Hadoop clusters secured with TLS/SSL.

In Hadoop, individual Hadoop services can be secured with TLS/SSL so that their communication with other secured services is encrypted. BDD can be configured to work with secured HDFS, YARN, Hive, and Key Management Server (KMS) at install or upgrade time. This results in its communications with those services being encrypted. For more information, see the *Installation Guide*.

Kerberos

The Kerberos network authentication protocol enables client/server applications to identify one another in a secure manner, even when communicating over an unsecured network.

Individual applications are called *principals* in Kerberos terminology. Each principal has a *keytab file*, which contains its *key*, or password. When one principal wants to communicate with another, it presents its keytab file for authentication and is only granted access to the other principal if its name and key are recognized. Because keytab files are protected using strong encryption, this process still works over unsecured networks.

BDD can be configured to use Kerberos authentication for its communications with Hadoop. You can set this up before or after installation; for more information, see the *Installation Guide* and the *Administrator's Guide*.

WebLogic Server security

WebLogic Server provides a J2EE container for hosting and managing Studio and the Dgraph Gateway Java applications. Additionally, WebLogic's Admin Server is used to perform many administrative tasks for the BDD cluster.

To help keep WebLogic Server secure, it is automatically installed with the minimum number of packages required by BDD. To further improve its security:

- Install it in production mode, which sets it to run with settings that are more secure and appropriate for a production environment.
- Never run it in development mode within a production environment.
- Replace its SSL certificated after installing.

For more information, see the *Installation Guide*.

Connection filters

WebLogic Server also supports domain connection filters, which enable you to deny access to domains at the network level. These are similar to firewalls in that they can be configured to filter on protocols, IP addresses, and DNS node names, and can protect server resources on individual servers, clusters, or an entire internal network. For example, you could set them to deny all non-SSL connections originating from outside your corporate network.

For more information on connection filters, see *Using Network Connection Filters*.

Further reading

More information on WebLogic security is available in the following guides:

- *Understanding WebLogic Server Security*
- *Administering Security for Oracle WebLogic Server*
- *Securing a Production Environment*
- *Securing Resources Using Roles and Policies*

Jetty security

Jetty provides an open-source `javax.servlet` container for hosting the BDD Transform Service, which is a RESTful service that enables Studio users to preview the effects of transformations on their data.

Jetty supports the Java Authentication and Authorization Service (JAAS), which is a collection of APIs used to identify users or other entities attempting to access a computer or service. This support is leveraged to provide basic authentication and authorization for the Transform Service's communications with other BDD components. Each Jetty instance within BDD includes a JAAS module, which automatically authorizes the credentials of components attempting to access the Transform Service.

More information on Jetty's JAAS support is available in the *Jetty documentation*.

Chapter 4

Data Set Security

This section describes options for securing your data sets.

Sentry

Sentry is a Hadoop component that provides role-based authorization in a Hadoop cluster. Among other things, it can be used to restrict access to Hive data at a granular level.

Oracle strongly recommends using Sentry to protect your data from outside users. For more information on using BDD in a Sentry-enabled cluster, see the *Installation Guide*.

HDFS data at rest encryption

HDFS data at rest encryption allows HDFS data to be stored in encrypted directories called *encryption zones*. All files within an encryption zone are transparently encrypted and decrypted on the client side, meaning decrypted data is never stored in HDFS.

You can enable this in your Hadoop cluster and then configure BDD to store its Dgraph databases, Avro files, and other data in encryption zones. This ensures that your BDD data will be safe even if HDFS is compromised.

Important: Enabling HDFS data at rest encryption for BDD only means that your data will be encrypted *while it's stored on HDFS*. Files are automatically decrypted when BDD components read them and re-encrypted when they're written back to HDFS, but they *aren't* encrypted while they're being handled by BDD components.

For more information on configuring HDFS data at rest encryption for BDD, see the *Installation Guide*.

Data set whitelists and blacklists

BDD whitelists and blacklists control which Hive tables are processed by the DP CLI. Whitelists specify tables that should be processed and blacklists specify tables that should be ignored. Only tables that have been processed are available to BDD and its users.

BDD provides default lists in `$BDD_HOME/dataprocessing/edp_cli/config/`. By default, the blacklist excludes all tables and the whitelist is empty, so you don't need to worry about accidentally processing sensitive data immediately after installation. No data will be processed until you modify these files, even if the Hive Table Detector is configured to run automatically.

The blacklist overrides the whitelist—if a file is specified in both lists, it won't be processed. Additionally, **the lists only work when they're used with the DP CLI**.

To keep your sensitive data secure, be sure to specify any tables you *don't* want processed in the blacklist, and *always* include this file when launching Data Processing jobs. If the Hive Table Detector is scheduled to run automatically, verify that the blacklist is specified in the `crontab` file (if you enabled the Hive Table Detector at install time, this should already be set). Also, as previously mentioned, access to the DP CLI and its whitelist and blacklist should be restricted.

For more information on using whitelists and blacklists, see the *Data Processing Guide*.

User roles

Studio users are assigned application-wide roles that determine which parts of the application they can access and the data they can view. Studio administrators can assign and modify user roles.

There are four types of user roles:

Role	Description
Administrator	Administrators have full access to all projects and data sets within Studio and can export data sets to HDFS to make them available to other users. They can also access Studio's Control Panel, where they can configure Studio settings and manage users. Because administrators have so much power, you should only assign this role to users who absolutely need it and are known to be trusted.
Power User	Power users can create projects, add and view data sets, edit their account information, and export data sets to HDFS. They can also access any projects they're assigned to, although their level of access depends on their role within each. Their access to data sets is limited by data set permissions. Since this role can add data to HDFS, you should make sure users can be trusted before assigning it to them.

Role	Description
User	Users can create projects, add and view data sets, and edit their account information. They can also access projects they're assigned to, although whether they have read/write or read-only access depends on their role within each. Their access to data sets is limited by data set permissions. Unlike power users, they can't export data sets to HDFS. Most of your users will have this role. Because it has reasonably limited access (no access to Studio configuration or HDFS), it's appropriate to assign to a new user.
Restricted User	Restricted users have read-only access to projects they're given a role in. They can't create or edit projects, or edit their account information. They're also unable to view data sets, although they can view the data in their assigned projects. Because this role has very limited access, it's appropriate to assign to a new or temporary user.

For more information on user roles, see the *Administrator's Guide* and the *Studio User's Guide*.

Inherited roles

A Studio user might have a number of assigned roles. In addition to a user role, they may have a project-specific role and belong to a user group that grants additional roles. In these cases, the highest privileges apply to each area of Studio, regardless of whether they were assigned directly or inherited from a user group.

Project roles

In Studio, project roles determine which users can access project data and configuration.

For a given project, users can have one of two roles:

- **Project author:** The project author can configure and manage the project, add and remove users and user groups, transform project data, and perform other similar tasks. This role is automatically assigned to the user who created the project, and can later be given to other users. Administrators have this role for all projects.
- **Project restricted user:** A project restricted user can view the project and its pages, as well as add and configure project pages and components.

Project roles can be assigned and managed by Studio administrators and project authors.

For more information on project roles, see the *Studio User's Guide* and the *Administrator's Guide*.

Project types

In Studio, a project's type determines which users can access it.

There are three project types:

Project type	Description
Private	Private projects can only be accessed by the project creator and Studio administrators, and their **All Big Data Discovery users** group is set to **No Access**. All projects are private by default; access must be granted by either the creator or an administrator.
Public	A project becomes public when its **All Big Data Discovery users** group is set to **Project Authors**.
Shared	A project becomes shared when: • Users other than the creator are added to it • User groups other than **All Big Data Discovery admins** and **All Big Data Discovery users** are added to it • The **All Big Data Discovery users** group is set to **Project Restricted Users**

For more information on project authors and project restricted users, see *Project roles on page 14*. For more information on project types, see the *Studio User's Guide*.

Data set permissions

Default access to data sets in Studio is determined by how they're created.

Initially, default data set access is as follows:

- Data sets created by users personally uploading files are private. Only the users who uploaded the files and administrators can access them. However, those users and administrators can grant read-only or read/write access to other users and groups.
- Data sets created by ingesting data from Hive are public.
- Data sets created by duplicating or exporting data are public, although this default setting can be changed by Studio administrators.

Once a data set has been created, any user with write access to it can change its permissions. For more information, see the *Studio User's Guide*.

Custom transform scripts and visualizations

Studio enables users to create custom visualizations and transform scripts, which require them to do some coding. In these cases, special security measures have been implemented to prevent users from accessing sensitive data.

Transform scripts

Custom transform scripts are Groovy scripts that users write and run in Studio's Transform component. For security purposes, Transform only supports a subset of the Groovy language, which excludes functions that could be used to compromise your data or your system. Scripts containing these functions will produce an error. For more information, see the *Studio User's Guide*.

Visualizations

The Custom Visualization Component (CVC) is an extension to Studio that lets you create customized visualizations in cases where the default Studio components don't meet your specific needs. Custom visualizations obtain the data they display by executing hard-coded EQL queries.

To prevent users from using them to access sensitive data, these queries are parsed and validated at runtime to ensure they're compliant with Studio's security restrictions. For more information on the CVC, see the *Extensions Guide*. More information on EQL is available in the *EQL Reference*.

Security Manager

The Security Manager is a Studio feature that filters the data that users can see in Studio. If you want to provide extra security for Studio, you can use the Component SDK to create a custom Security Manager to restrict access to specific data.

The Component SDK package contains a sample Security Manager that you can use as a starting point. You can configure the Security Manager to filter your data at the record level based on user, group, and role. Once your Security Manager is complete, you can deploy it and configure Studio to use it.

Note: Record-level security enforced by the Security Manager doesn't apply to your data at the Hadoop level.

For more information on the Security Manager, see the *Studio User's Guide* and the *Extensions Guide*.

Chapter 5

Studio User Access

This section describes ways to control the way users access Studio.

Managing Studio users

In situations where LDAP isn't used, Studio administrators can add, edit, and remove users.

When adding a new user, you specify their details, including:

- Their password. You can also specify whether they're required to change it the first time they log in. To ensure the security of both the user and the application, this should always be required.
- Their role: administrator, power user, user, restricted user, or none. You should verify that the user can be trusted before making them an administrator or power user.
- Their level of data set access: read only or read/write.
- Their access to individual projects and their role within each. In scenarios that require strict security, new uses should only have access to projects they need to be involved in.

Once the user has been added, administrators can edit the above information at any time. They can also edit roles the user inherits from any groups they belong to.

Administrators also have the option of deactivating users. Once a user has been deactivated, they can be either reactivated or deleted.

For more information on managing users, see the *Administrator's Guide*.

User authentication

In situations where LDAP isn't used, administrators can configure user authentication with Studio's own user management tools.

Studio users typically log in with their email addresses, but you can enable them to use their screen names instead. You can also restrict certain email addresses and screen names, and enable users to log in automatically.

Additionally, Studio provides tools for creating and enforcing password policies. For example, you can:

- Enable users to change their own passwords.

- Require users to change their passwords the first time they log in.
- Prohibit the use of dictionary words in passwords.
- Specify a minimum password length.
- Prohibit users from reusing a certain number of their most recent passwords.
- Set passwords to expire within a certain time frame.

To keep users and the application itself secure, you should create a strong password policy. At a minimum, Oracle recommends you require passwords to contain at least six characters and set them to expire periodically.

For more information on configuring user authentication, see the *Administrator's Guide*.

LDAP

Studio can be integrated with an existing Lightweight Directory Access Protocol (LDAP) system. This is supported for systems that use TLS/SSL and those that don't.

When this is enabled, Studio accounts are automatically added for users in LDAP groups that are configured to work with BDD. Those users can then use their LDAP credentials to sign in to Studio. Note that when LDAP is enabled, you can still add users through Studio.

By default, Studio uses the passwords and password policy defined in the LDAP system. This reduces the number of passwords users need to keep track of and relieves administrators of the responsibility of creating and maintaining password policies. You can, however, configure Studio to enforce its own policy, if necessary.

Studio automatically stores an encrypted version of each user's LDAP password so that users can log in when Studio can't connect to the LDAP server. Although this feature is convenient, it can be disabled in scenarios that require stricter security.

You can enable this from within Studio. Note that systems that use TLS/SSL require extra configuration. For more information, see the *Administrator's Guide*.

SSO

You can integrate Studio with single sign-on (SSO) to enable users to log in through the SSO system.

When SSO is enabled, all user accounts are managed by the SSO system—you can't create or edit users in Studio. This means that you can't use the default administrative user created during installation. You should therefore ensure that at least one SSO user has an administrative role for BDD beforehand.

The officially supported method of enabling SSO is with Oracle Access Manager and an Oracle HTTP Server in front of BDD. This process is described in the *Administrator's Guide*.