# TIBCO® Data Virtualization

## Security Features Guide

Version 8.8.0 | October 2023

# Contents

# About TDV and BD Security Features

The TIBCO® Data Virtualization (TDV) and Business Directory (BD) have many security features and design considerations engineered to work together as an integrated system. These features keep information secure and available for use by only authenticated and authorized individuals with the appropriate rights and privileges.

Because TDV and BD are installed together and work in concert, references to TDV security apply to BD as well.

- TDV installer security during silent and interactive installations
- Security domains
- Internal repository security
- Log file security
- Encryption
- Export files security
- Cached data security
- Monitor security
- JDBC, ODBC, and ADO.NET TDV client security
- Manager client security
- Studio client security
- Web service client security
- Web service security standards
- Legacy web service security standards (transport layer and data source security)
- Security for the “composite” domain

For a discussion of user rights and privileges, refer to the *TDV Administration Guide*.

**Note:** This document lists the supported security features, but does not describe them in detail or explain how to set them up. Details and setup instructions can be found in other TDV and BD manuals. Each manual that contains security information has a centralized cross-reference list in its introductory chapter.

# Security Feature Highlights

The following are the highlights of TDV security features:

- Passwords sent by JDBC and ODBC to TDV are encrypted.
- Passwords passed between TDV components are encrypted.
- Passwords in HTTP/SOAP headers for administrative functions are encrypted.
- All communication between TDV and Studio can be encrypted using SSL/HTTPS. The default TLS version will default to the highest supported version that is negotiated between TDV and Studio (Examples: Java 8 defaults to TLS v1.2 and Java 11 defaults to TLS v1.3).
- WSS Web service client security is supported.
- TDV to data source SSL is supported with or without Web Service client authentication where permitted.
- Passwords in metadata are encrypted.
- Passwords for LDAP and dynamic domain users are encrypted or not stored.
- Support for case-sensitive user login for external LDAP is supported.
- Options to include or exclude encrypted user passwords, repository passwords, LDAP, and data source passwords in export files.
- DBA password for the repository is not stored.
- Repository password and the repository connection with TDV are encrypted.
- Passwords are not shown in the log files.

For information on how administrators can delegate administrator rights, add LDAP users to a TDV group, and grant/revoke access privileges on TDV resources, see the *TDV Administration Guide.*

# TDV Security Features by Component

The TIBCO® Data Virtualization (TDV) forms the core of the Data Virtualization Platform. You can use this document to evaluate TDV security features.

Topics covered in this chapter:

# TDV Installer Security

TDV supports security for both silent and interactive installation.

TDV installers running in the UI or console mode do not store clear text passwords in any log files. The database administrator password for the repository is not stored.

The TDV installer installs a PostgreSQL database.

## Configuring Security Enhanced Linux Environments

All Redhat OS Linux variants that have SELinux support can utilize it enabled (i.e. SELinux = enabled). If one wants to run TDV and/or Business Directory with SELinux enabled then an appropriate security policy that allows read/write access to the TDV installation directory and TDV ports is required before installation.

SELinux configuration file is located under /etc/selinux/config.

To configure SE Linux environments

1. Login as root on your Linux instance.

2. Run /usr/sbin/sestatus to validate your setting for SELinux.

3. If SELinux=enabled then you will need to make sure the following is part of your security policy.

a. TDV installation directory (TDV and/or Business Directory) must have read and write privileges on that directory and all files/directories underneath.

b. TDV ports (refer to Installation Guide "Port Requirements" section regarding what ports to allow)

# Script output:Repository Security

This section describes TDV security for its repository when it is an internal implementation.

The password for the repository connection is encrypted.

For a brief time after initial installation, the repository connection password is in clear text. The first time TDV starts after initial installation, if it detects that the repository connection password is in clear text, it encrypts the password and stores it as a file name and key.

- File name: <TDV_install_dir>/conf/server/server_values.xml
- Key: /server/config/database/databasePassword

Administrator passwords for LDAP domains are encrypted. The database admin password appears in cis.repository while the installer_services.sh script is running. When the script is completed, the password is automatically deleted from cis.repository.

# Repository and Cache Database Access and Privileges

The installer creates the repository and cache databases, it sets access and privileges for root, cisrepo and tutorial users.

The root user account is the DBA account for the entire PostgreSQL database.

The cisrepo user account is used to manage the TDV Server repository and cache databases.

The tutorial user account is used to manage the example Postgresql database used by the TDV Server for demo purposes.

## Remote Access to the Repository

No user can access the repository nor the cache database remotely.

It is configured to only allow local connections.

## Repository Settings

Repository settings are as shown in the following table.

| Repository DB Name |
| --- |
| |
| |

| Repository DB Name |
| --- |
| |
| |
| |
| |

## Cache Settings

Cache database settings are as shown in the table.

| Cache DB Name |
| --- |
| |
| |
| |
| |

Data cached to relational databases is stored in clear text.

If flat file caching is used, the data is saved in binary format on the TDV installation drive and rendered in clear text. Secure access to HOMEDRIVE/temp or the customized location can be configured for file caching.

# Log File Security

Passwords that occur in log files generated by TDV and clients are obfuscated.

# Encryption

This section describes TDV password encryption.

TDV uses AES and Tiny Encryption Algorithm Variant (TEAV) for password encryption. For details on TEAV, visit: http://www.axlradius.com/freestuff/TEAV.java.

The JDBC driver supports RSA encryption. Each JDBC connection uses a unique RC4 session key to encode the users password for transport. The ADO.NET and ODBC drivers still use TEAV.

The TDV stores password hashes or encrypted passwords for users of the composite domain, but does not store passwords for LDAP or pass-through users.

## TDV Command Line Utilities

The following TDV command line utilities make use of a -optfile or -configFile option, which can be used to hide password information from the command line:

- pkg_import.<sh|bat>
- pkg_export.<sh|bat>
- backup_import.<sh|bat>
- backup_export.<sh|bat>
- repo_util.<sh|bat>

# Studio Session Security

For non-SSL session protection you can use the Studio Session Authentication configuration parameter. This configuration parameter adds an extra authentication level for session protection between the TDV Server and Studio for use with unencrypted connections. This parameter can be found in the TDV Studio client (goto Administration->Configuration and search for the parameter “studio session”) Navigate to the property Server->Configuration->Security->Authentication->Studio Session Authentication. Following are the values for this property. The default value is False.

- True—use session protection.
- False—don't use session protection.

# Data Source Security

The following table describes how security is maintained between TDV and data sources.

| Descriptions | 8.0 and Above |
|---|---|
| Data source passwords encrypted and stored in the TDV metadata repository using: | AES |
| The TDV Server passes connection profile information to a vendor-supplied database driver, which encodes login and password according to vendor specifications and negotiates a secured session connection between the targeted database and the TDV Server. | Database driver encoding |
| Passwords in HTTP / SOAP headers during data source Web Service invocations are sent in the following formats: | Clear text, base64-encoded |
| Web Service security for data sources support. | Supported |
| Pass-through of incoming non-standard HTTP headers to data sources over HTTP. | Configured per header per data source |
| Web Service: NTLM authentication through an NTLM header. | Supported |
| Web Service: NTLM authentication through a Negotiate header. | Not supported |
| Web Service: Kerberos authentication through a Negotiate header. | Supported |
| Delegation/forwarding of client credentials to Kerberos data sources. | Supported for Sybase and Oracle thin drivers |
| Kerberos access to Sybase databases. | Supported |
| Kerberos access to MS SQL Server databases. | Supported. Not supported for data ship. |

| Descriptions | 8.0 and Above |
|---|---|
| Kerberos access to Greenplum databases. | Supported |
| Kerberos access to Oracle databases. | 11g drivers to 11g and 10g databases |

# Export Files Security

This section describes TDV password security for export files.

By default, data source passwords are excluded from the package export (CAR) files. When explicitly included, they are encrypted.

Data source passwords in package export directories are encrypted.

User passwords for users in the composite domain are encrypted and included in full server backup export (CAR) files, but they are not included by default in package export files. When users are included in the export file, the passwords are encrypted.

The repository password is included in full server backup export files, but not in package export files. When included, the repository password is encrypted.

The password for each LDAP domain (but not passwords for LDAP users) is included in full server backup export files, but not in package exports. When included, LDAP domain passwords are encrypted.

# Monitor Security

Passwords in HTTP or SOAP headers for the following listed actions are encrypted using base64 encoding:

- Flush repository cache
- Get server list
- Get server status
- Stop monitor
- Start server

- Stop server
- Restart server

# JDBC, ODBC, and ADO.NET Client Security

TDV provides the following security for communications with clients via JDBC or ODBC or ADO.Net..

| Driver and Path | When | Encryption |
|---|---|---|
| JDBC programs to TDV | Performing TDV authentication during Create Connection process | RSA |
| JDBC programs to and from TDV, throughout connection | encrypt=true | Data encoded with TLS/SSL |
| JDBC programs to and from TDV | During connection | User password encoded with unique RC4 session key |
| ODBC programs to TDV | Performing TDV authentication during Create Connection process | TEAV |
| ADO.NET, throughout connection | During connection | TEAV |

TDV supports the following Single Sign-On access:

| Driver | SSO Access Through |
|---|---|
| JDBC | Kerberos |
| ODBC | Kerberos, NTLM |

| Driver | SSO Access Through |
| --- | --- |
| ADO.NET | Kerberos, NTLM |

# Manager Clients Security

Initial LDAP domain creation and update sends login and password from Manager to TDV as clear text.

Create and update LDAP domain connection profiles using a browser launched locally on the TDV installation.

# Studio Client Security

Passwords sent from Studio to TDV during user authentication are encrypted.

Passwords sent between Studio and TDV during data source create or update processes are encrypted.

Passwords sent between Studio and TDV during domain create and update processes are not supported. Domain, group, and user management can be done using TDV Manager.

Passwords in HTTP/SOAP headers for the following actions are encrypted and base64-encoded:

- Flush Repository Cache
- Get Server List
- Get Server Status
- Start Server
- Stop Server
- Restart Server
- Fetch Logs

Single Sign-On access using Kerberos is supported.

# TDV Server Security

In order to provide security from the host header attack, TDV provides a configuration option that users can tune. Navigate to the property Server-> Configuration-> Security-> Allowed Hosts. Sites mentioned in this list determines the allowed host/domain names.

A fake Host value in incoming HTTP request headers can be used for Cross-Site Request Forgery, cache poisoning attacks, and poisoning links in emails. This configuration determines the allowed host/domain names.

Values in this list can be fully qualified names (e.g. 'www.example.com'), in which case they will be matched against the request's Host header exactly (case-insensitive, not including port). A value beginning with a period can be used as a subdomain wildcard: '.example.com' will match example.com, www.example.com, and any other subdomain of example.com.

Default value is empty which means the Host header is not validated.

Changing this value will have no effect until the next server restart.

# Kerberos Support

Enterprise users can leverage Kerberos infrastructure to authenticate just once to secure access to TDV-defined resources. The duration of an authenticated session is set by the Kerberos administrator. TDV supports pass-through of the Kerberos tokens from the authenticated client through TDV to the Kerberos server and to the data sources. The TIBCO® Data Virtualization Server, data sources, and clients of the TDV Server must be configured to support Kerberos token pass-through and SSO.

The KDC Kerberos v5 Server must already be installed and running in your environment before you install TDV Server and Studio. You then configure the Kerberos system to use with TDV, establishing a security context in which Kerberos and the TDV identify each other.

# Disabling the Drill Console

The TDV Server embeds Apache Drill to assist with the runtime for the MPP Engine. The Apache Drill console uses an older version of jQuery which has known security vulnerabilities. The Apache Drill console is not required for normal operation of TDV or its

MPP capability. This console can safely be disabled by blocking access to it via OS configuration (for on-premise installations), Docker Container/Pod configuration, or Kubernetes Helm Chart.

**TDV Server (on-premises):**

On operating system hosting TDV Server (Windows/Linux/AIX), change the firewall to not expose (TDV_BASE_PORT - 100)+3.

Example: TDV Server installation on Linux using default TDV_BASE_PORT=9400.

Change firewalld or iptables to not allow inbound requests to port 9303.

**TDV Server (docker) :**

Modify TDV container/pod to not expose port (TDV_BASE_PORT - 100) + 3.

Example: Change the TDV Docker run_tdv_container.sh script to only allow external access to 9300-9302,9304-9306 using the content below.

*Assumption: default TDV_BASE_PORT=9400. All previous ports will continue to be open except (TDV_BASE_PORT - 100) + 3.*

| Old content | New content |
|---|---|
| `local H_DRILL_PORT_1=$(($H_DRILL_BASE_PORT + 1)); local C_DRILL_PORT_1=$(($C_DRILL_BASE_PORT + 1))` | `local H_DRILL_PORT_2=$(($H_DRILL_BASE_PORT + 2)); local C_DRILL_PORT_2=$(($C_DRILL_BASE_PORT + 2))` |
| | *`Add this new line`*<br>`local H_DRILL_PORT_4=$(($H_DRILL_BASE_PORT + 4)); local C_DRILL_PORT_42=$(($C_DRILL_BASE_PORT + 4))` |
| `CONTAINER_PORTS="-p ${H_BASE_PORT}-${H_PORT_3}:${C_BASE_PORT}-${C_PORT_3} -p $H_PORT_5:$C_PORT_5 -p $H_PORT_9:$C_PORT_9 -p ${H_DRILL_BASE_PORT}-${H_DRILL_PORT_2}:${C_DRILL_BASE_PORT}-${C_DRILL_PORT_2} -p ${H_DRILL_PORT_4}:${C_DRILL_PORT_6}"` | `CONTAINER_PORTS="-p ${H_BASE_PORT}-${H_PORT_3}:${C_BASE_PORT}-${C_PORT_3} -p $H_PORT_5:$C_PORT_5 -p $H_PORT_9:$C_PORT_9 -p ${H_DRILL_BASE_PORT}-${H_DRILL_PORT_6}:${C_DRILL_BASE_PORT}-${C_DRILL_PORT_6}"` |

**Note**: If you are not running the *run_tdv_container.sh* script that is provided with TDV, then you must manually exclude port 9303 when running the container using the ***docker run*** command. For example:

```
docker run -itd -v <volume>:<location> type=volume,source=tdv-
vol,target=/opt/TIBCO --cpus=2.000 -env [TDV_ADMIN_PASSWORD=<PASSWORD>]
[TDV_ADMIN_PASSWORD_FILE=<FILE with tdv admin password>] -m=8g -p
9300:9300 -p 9301:9301 -p 9302:9302 -p9304:9304 -p9305:9305 -p 9306:9306
-p 9400:9400 -p 9401:9401 -p 9402:9402 -p 9403:9403 --hostname=localhost
--name tdv myrepo/tdv:8.4 Dockerfile.tdv.repo
```

**TDV Server Kubernetes:**

Modify TDV helmchart to not expose port (TDV_BASE_PORT - 100) + 3.

Example: The content of the TDV helmchart should be changed to only allow external access to 9300-9302,9304-9306 as described in the table below.

*Assumption: default TDV_BASE_PORT=9400. All previous ports will continue to be open except (TDV_BASE_PORT - 100) + 3.*

| tdv.yaml | tdv-svc.yaml |
|---|---|
| Remove the following content from the file *tdv/templates/tdv.yaml* | Remove the following content from the file *tdv/templates/tdv-svc.yaml* |
| `- name: "p9303"`<br>`containerPort: 9303` | `- port: 9303`<br>`name: "p9303"`<br><br>`- port: 9303`<br>`nodePort: 31303`<br>`name: "p9303"`<br><br>`- port: 9303`<br>`name: "p9303"` |

# Web Service Client Security

TDV supports a variety of Web Service security standards. These are listed and explained in the following sections:

- Supported Web Service Security Standards

- Explanation of Web Services Policy

# Supported Web Service Security Standards

TDV supports the following Web Service client security standards:

- Passwords in HTTP / SOAP headers during Web Service invocations to or from TDV Server in clear text, base64-encoded
- WS-Security for Web Service clients (next section)
- WSSE UsernameToken SOAP headers, used instead of transmitting usernames and passwords (composite domain only). For this to work, the Store User Password configuration parameter must be changed to True from its default setting of False.
- X-WSSE UsernameToken HTTP extension header instead of transmitting usernames and passwords (composite domain only)
- Use of WSSE and X-WSSE authentication require the server to be configured to store passwords in the repository rather than hash values.
- NTLM authentication through an NTLM header
- NTLM authentication through a Negotiate header
- Kerberos authentication through a Negotiate header

Data source Web Service invocations from TDV Server can support SSL with or without client authentication (if the data source supports SSL).

The following security policies, in the form of XML files, are provided for Web Service clients.

| Transport or Standard | System Security Policy | Description |
|---|---|---|
| HTTP | Http-Basic-Authentication.xml | Policy that requires a user name and password when making a request. |
| HTTP | Http-UsernameToken-Digest.xml | Policy that validates against a UsernameToken header encrypted using a nonce value. |

| Transport or Standard | System Security Policy | Description |
|---|---|---|
| HTTP | Http-UsernameToken-Plain.xml | Policy that validates against a UsernameToken header. The password can be in plain text. |
| HTTPS | Https-Basic-Authentication.xml | Policy that requires a user name and password when making a request. |
| HTTPS | Https-ClientCertificateRequire.xml | Policy that requires client certificates. |
| HTTPS | Https-UsernameToken-Digest.xml | Policy that validates against a UsernameToken header encrypted using a nonce value. |
| HTTPS | Https-UsernameToken-Plain.xml | Policy that validates against a UsernameToken header. The password can be in plain text. |
| SOAP | UsernameToken-Digest.xml | Policy that validates against a UsernameToken header encrypted using a nonce value. |
| SOAP | UsernameToken-PlainText.xml | Policy that validates against a UsernameToken header. The password can be in plain text. |
| SAML | Saml1.1-Bearer-Wss1.1.xml | Method in which the bearer assertion is used to facilitate single sign-on to the web browser. |
| SAML | Saml1.1-HolderOfKey-Wss1.0.xml | Method that establishes a correspondence between a SOAP message and the SAML assertions added to the SOAP message. |
| SAML | Saml1.1-SenderVouches- | Subject-confirmation method that |

| Transport or Standard | System Security Policy | Description |
|---|---|---|
| | Wss1.1.xml | enables an attesting entity to vouch for the identity of a subject to a party that trusts the sender. |

# Explanation of Web Services Policy

A Web Services policy is the same as an authentication scheme, but it is expressed in the form of an element in an XML file.

The XML snippet is referred to as a Web Services policy. The format of the XML snippet is described in Web Services Policy 1.2 - Framework (WS-Policy), which is at:

http://www.w3.org/Submission/WS-Policy/

This is the official specification, and the best source of reference for understanding policies.

Policies are server-side. The reason for selecting a policy is to tell the TDV server what authentication scheme to use when authenticating requests from clients.

TDV uses the Metro JDK to implement the authentication for each policy. Metro is documented at:

http://www.ibm.com/developerworks/java/library/j-jws10/index.html

Web policy basics are explained at:

http://www.ibm.com/developerworks/java/library/j-jws18/

Further information about Web Service policy can be found at:

- TDV Administrator Security Policy Actions
- SOAP Web Service Example
- REST Web Service Example

## TDV Administrator Security Policy Actions

As TDV administrator, you need to select the policy that represents the desired authentication scheme. You do *not* need to:

- Modify the policies (for example, edit the XML policy element)

- Do anything except select a policy for the server side
- Do anything at all on the client side

### SOAP Web Service Example

To select Basic Authentication for a SOAP Web Service, select the policy that represents the authentication scheme from the Security Policy drop-down list in the Service pane on the SOAP panel for the Web Service you are publishing. For a description of the procedure, see “Publishing a WSDL SOAP Data Service” or “Publishing a Contract-First WSDL SOAP Data Service” in the *TDV User Guide*.

### REST Web Service Example

To select Basic Authentication for a REST Web Service, make sure Enable HTTP Basic is set to true in the Service pane on the REST panel for the Web Service you are publishing. For a description of the procedure, see “Publishing a WSDL REST Data Service” in the *TDV User Guide*.

# Authentication between Clients and TDV

The table below lists the authentication protocols supported between clients and the TDV Server.

| Authentication Protocol | TDV Support |
|---|---|
| Kerberos | Active |
| LDAP | Active |
| Kerberos for LDAP | Active |
| NTLM | Active |

# Using Kerberos Constrained Delegation

TDV JDBC driver can also be configured to use Kerberos Constrained Delegation. This feature allows a service to obtain service tickets to a restricted list of other services running on specific servers on the network after it has been presented with a service ticket. For more details on the process see: https://technet.microsoft.com/en-ca/library/cc995228.aspx.

The userGSSCredential connection property can be used in the connection URL to pass in a GSSCredential object. The following sample code shows how to use the property to pass the GSSCredential into the driver using JDBC:

```
GSSCredential impersonatedUserCredential = [userCredential]
```

```
Properties driverProperties = new Properties();
```

```
Driver driver = (Driver) Class.forName
("cs.jdbc.driver.CompositeDriver").newInstance();
```

```
driverProperties.setProperty("authenticationMethod", "kerberos");
```

```
driverProperties.put("userGSSCredential", impersonatedUserCredential);
```

```
Connection conn = DriverManager.getConnection(CONNECTION_URL,
driverProperties);
```

```
GSSCredential impersonatedUserCredential = [userCredential]
```

```
CompositeDataSource datasource = new CompositeDataSource();
```

```
datasource.setURL(CONNECTION_URL);
```

```
datasource.setUserGSSCredential(impersonatedUserCredential);
```

```
Connection conn = datasource.getConnection();
```

# Authentication between TDV and Data Sources

The table below lists the authentication protocols supported between the TDV Server and data sources.

| Authentication Protocol | TDV Support |
|---|---|
| Kerberos for IBM DB2 LUW v9.5 | Active |
| Kerberos for MS SQL Server 2008 and 2012 | Active |
| Kerberos for HiveServer2 | Active |
| Kerberos for Oracle 10g | Active |
| Kerberos for Oracle 11g | Active |
| Kerberos for SOAP | Active |
| Kerberos for REST | Active |
| Kerberos for Sybase ASE v15 | Active |
| Kerberos for WSDL | Active |
| Kerberos for XML over HTTP | Active |
| NTLM for REST | Active |
| NTLM for SOAP | Active |
| NTLM for WSDL | Active |
| NTLM for XML over HTTP | Active |

# Composite Domain Security

TDV supports its own security domain, and one can define users and groups in it. However, these users and groups do not exist outside the TDV environment. Many organizations use Microsoft Active Directory or an LDAP server to manage users and groups throughout the enterprise. TDV allows users to introspect those Active Directory and LDAP servers, and create security domains inside TDV for them.

Because the composite domain does not exist outside TDV, user passwords are either hashed or encrypted, and stored in the security_members table in the TDV metadata repository. Also note the following:

- TDV stores passwords for each user in the composite domain.
- TDV does not store passwords for Active Directory and LDAP domains. Instead, TDV forwards user credentials to the Active Directory server for user authentication.
- TDV does not store passwords for dynamic domain users.
- TDV supports case-sensitive user logins from external LDAP domains.
- TDV does not allow implicitly anonymous LDAP login through blank passwords.

# SSL Protocol Configuration

In versions prior to TDV 7.0.8, TDV and Business Directory were configured to use only TLSv1 as the SSL protocol.

TDV 8.2 and above now use Java 11 which defaults to TLS v1.3.

If you want to make additional restrictions for SSL protocols then follow the next steps for configuration.

## TDV

1. Start TDV Server and launch the TDV Studio.
2. In the TDV Studio client, open the Administration > Configuration dialog and search for "Disabled Protocols for SSL Connectors”.
3. The default TDV value for “Disabled Protocols for SSL Connectors” is “SSLv2Hello, SSLv2, SSLv3”. Other valid values are “TLSv1, TLSv1.1, TLSv1.2, TLSv1.3, DTLSv1.0, DTLSv1.2”.

4. Restart the TDV Server.

### TDV Business Directory:

1. Start Business Directory server.
2. To make changes on server configuration, you may use rest api with your own tool. Using CURL, the call will look like:

```
curl -X PUT -u admin:<Business Directory admin password>
"https://<Business Directory IP>:9502/rest/v2/configs/_server_
communications_sslProtocolsToRemove" -d "value=SSLv2Hello,SSLv2,SSLv3"
```

**Note**: the api document is available at https://<TDV Business Directory IP>:9502/directory/api-docs/#!/configs/setValue

3. The default TDV value for “Disabled Protocols for SSL Connectors” is “SSLv2Hello, SSLv2, SSLv3”. Other valid values are “TLSv1, TLSv1.1, TLSv1.2, TLSv1.3, DTLSv1.0, DTLSv1.2”.
4. Restart the Business Directory server.

# Java Supported Cipher Suites

TDV and Business Directory use the Java Software for providing the cipher suite functionality. The following lists show the support cipher suites based on the OS platform:

### Linux x64/Windows x64 (Java 11)

An asterisk (*) indicates that the cipher in enabled by default

* TLS_AES_128_GCM_SHA256

* TLS_AES_256_GCM_SHA384

* TLS_DHE_DSS_WITH_AES_128_CBC_SHA

* TLS_DHE_DSS_WITH_AES_128_CBC_SHA256

* TLS_DHE_DSS_WITH_AES_128_GCM_SHA256

* TLS_DHE_DSS_WITH_AES_256_CBC_SHA

* TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
* TLS_DHE_DSS_WITH_AES_256_GCM_SHA384
* TLS_DHE_RSA_WITH_AES_128_CBC_SHA
* TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
* TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
* TLS_DHE_RSA_WITH_AES_256_CBC_SHA
* TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
* TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
* TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
* TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
* TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
* TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
* TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
* TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
* TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
* TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
* TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
* TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
* TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA
* TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256
* TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
* TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA
* TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384
* TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384
* TLS_ECDH_RSA_WITH_AES_128_CBC_SHA

* TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256
* TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256
* TLS_ECDH_RSA_WITH_AES_256_CBC_SHA
* TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384
* TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384
* TLS_EMPTY_RENEGOTIATION_INFO_SCSV
* TLS_RSA_WITH_AES_128_CBC_SHA
* TLS_RSA_WITH_AES_128_CBC_SHA256
* TLS_RSA_WITH_AES_128_GCM_SHA256
* TLS_RSA_WITH_AES_256_CBC_SHA
* TLS_RSA_WITH_AES_256_CBC_SHA256
* TLS_RSA_WITH_AES_256_GCM_SHA384

## AIX ppc64 (Java 11)

* TLS_AES_128_GCM_SHA256
* TLS_AES_256_GCM_SHA384
* TLS_DHE_DSS_WITH_AES_128_CBC_SHA
* TLS_DHE_DSS_WITH_AES_128_CBC_SHA256
* TLS_DHE_DSS_WITH_AES_128_GCM_SHA256
* TLS_DHE_DSS_WITH_AES_256_CBC_SHA
* TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
* TLS_DHE_DSS_WITH_AES_256_GCM_SHA384
* TLS_DHE_RSA_WITH_AES_128_CBC_SHA
* TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
* TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
* TLS_DHE_RSA_WITH_AES_256_CBC_SHA
* TLS_DHE_RSA_WITH_AES_256_CBC_SHA256

* TLS_DHE_RSA_WITH_AES_256_GCM_SHA384

* TLS_EMPTY_RENEGOTIATION_INFO_SCSV

* TLS_RSA_WITH_AES_128_CBC_SHA

* TLS_RSA_WITH_AES_128_CBC_SHA256

* TLS_RSA_WITH_AES_128_GCM_SHA256

* TLS_RSA_WITH_AES_256_CBC_SHA

* TLS_RSA_WITH_AES_256_CBC_SHA256

* TLS_RSA_WITH_AES_256_GCM_SHA384

# How To Disable Specific Ciphers

If you want to disable specific ciphers you can modify the JDK as follows:

Edit <INSTALL_DIR>/jdk/conf/security/java.security

```
jdk.tls.disabledAlgorithms=<ALGORITHMS>
```

JVM-wide algorithm restrictions for SSL/TLS processing. It is possible to disallow certain algorithms or limit key sizes.

These settings are available since Java 1.7. For more information on algorithms and usage, see this link:

https://www.java.com/en/configure_crypto.html.

**Note**: In certain cases, you may want to enable an older version of TLS to allow data source connections to work as expected. In such cases, review the list of disabled ciphers and edit the list as needed. For example Java 11 has older TLS versions (TLS v1 and v1.1) disabled. However, for certain older versions of data sources to connect successfully, you will need to enable these versions and therefore remove these from the disabled list in the java.security settings file.

# Updating JDK and Security Property

Follow these steps to manually update JDK:

1. Windows and Linux users, download the new JDK from:

   Oracle JDK (requires Oracle contract/license)

   AIX users, download the new JDK from:

   OpenJDK

2. Stop TDV java processes
3. Backup the current JDK:

```
mv <TDV_INSTALL_DIR>/jdk <TDV_INSTALL_DIR>/jdk_old
```

4. Unzip/untar new JDK under <TDV_INSTALL_DIR>/jdk
5. Delete the following:

   AIX/LINUX:

```
<TDV_INSTALL_DIR>/jdk/lib/libsunec.so
```

   Windows:

```
<TDV_INSTALL_DIR>/jdk/bin/sunec.dll
```

6. Backup the default java security property file:

```
copy INSTALL_DIR/jdk/conf/security/java.security INSTALL_
DIR/jdk/conf/security/java.security.orig
```

7. The java security provider list is maintained in the file: <TDV_INSTALL_DIR>/jdk/conf/security/java.security

The content of this file is:

**Linux x64:**

```
security.provider.1=SUN
```

```
security.provider.2=SunRsaSign
```

```
security.provider.3=SunEC
```

```
security.provider.4=SunJSSE
```

```
security.provider.5=SunJCE
```

```
security.provider.6=SunJGSS
```

```
security.provider.7=SunSASL
```

```
security.provider.8=XMLDSig
```

```
security.provider.9=SunPCSC
```

```
security.provider.10=JdkLDAP
```

```
security.provider.11=JdkSASL
```

```
security.provider.12=SunPKCS11
```

**Windows x64:**

```
security.provider.1=SUN
```

```
security.provider.2=SunRsaSign
```

```
security.provider.3=SunEC
```

```
security.provider.4=SunJSSE
```

```
security.provider.5=SunJCE
```

```
security.provider.6=SunJGSS
```

```
security.provider.7=SunSASL
```

```
security.provider.8=XMLDSig
```

```
security.provider.9=SunPCSC
```

```
security.provider.10=JdkLDAP
```

```
security.provider.11=JdkSASL
```

```
security.provider.12=SunMSCAPI
```

```
security.provider.13=SunPKCS11
```

AIX ppc64:

```
security.provider.1=SUN
```

```
security.provider.2=SunRsaSign
```

```
security.provider.3=SunEC
```

```
security.provider.4=SunJSSE
```

```
security.provider.5=SunJCE
```

```
security.provider.6=SunJGSS
```

```
security.provider.7=SunSASL
```

```
security.provider.8=XMLDSig
```

```
security.provider.9=SunPCSC
```

```
security.provider.10=JdkLDAP
```

```
security.provider.11=JdkSASL
```

```
security.provider.12=SunPKCS11
```

Whenever, there is a change to the list of providers, the java.security file has to be updated, so that TDV security settings will be applied correctly.

Specifically, care should be taken that

a. whenever a provider is removed from this list, the list should be renumbered so as to maintain the correct sequence. Also make sure that the TDV security file mentioned in step 8 below is updated.

b. whenever a new provider is added, always add it to the end of the above list and make sure that the TDV security file mentioned in step 8 below is updated..

***Note***: Java's default security property has SunEC in the list, but it's not loaded since the module file (either .so or .dll) was removed in #5 above.

8. The two TDV security files are in the following locations:

- <TDV_INSTALL_DIR>/conf/server/security/java.security (all TDV Server supported platforms)
- <TDV_INSTALL_DIR>/apps/drill/conf/java.security (only for Linux x64 platforms).

The content of these files is:

**Linux x64**

```
security.provider.13=BC
```

**Windows x64**

```
security.provider.14=BC
```

### AIX ppc64

```
security.provider.13=BC
```

The sequence number mentioned here is one more than the last number mentioned in the JDK security file. Whenever the sequence number in the JDK security file changes, care must be taken to update the content of the TDV security files.

# TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

## How to Access TIBCO Documentation

Documentation for TIBCO products is available on the Product Documentation website, mainly in HTML and PDF formats.

The Product Documentation website is updated frequently and is more current than any other documentation included with the product.

## Product-Specific Documentation

The following documentation for this product is available on the TIBCO® Data Virtualization page.

**Users**

- TDV Getting Started Guide
- TDV User Guide
- TDV Web UI User Guide
- TDV Client Interfaces Guide
- TDV Tutorial Guide
- TDV Northbay Example

**Administration**

- TDV Installation and Upgrade Guide
- TDV Administration Guide
- TDV Active Cluster Guide
- TDV Security Features Guide

**Data Sources**

- TDV Adapter Guides

TDV Data Source Toolkit Guide (Formerly Extensibility Guide)

**References**

TDV Reference Guide

TDV Application Programming Interface Guide

**Other**

TDV Business Directory Guide

TDV Discovery Guide

TDV and Business Directory Release Notes - Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

## Release Version Support

TDV 8.5 and 8.8 are designated as Long Term Support (LTS) versions. Some release versions of TIBCO® Data Virtualization products are selected to be long-term support (LTS) versions. Defect corrections will typically be delivered in a new release version and as hotfixes or service packs to one or more LTS versions. See also Long Term Support.

## How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our product Support website.
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the our product Support website. If you do not have a username, you can request one by clicking **Register** on the website.

## How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature

requests from within the TIBCO Ideas Portal. For a free registration, go to TIBCO Community.

# Legal and Third-Party Notices

SOME CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.



TIBCO, TIBCO logo, TIBCO O logo, ActiveSpaces, Enterprise Messaging Service, Spotfire, TERR, S-PLUS, and S+ are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (https://www.cloud.com/legal) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: https://scripts.sil.org/OFL

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the “readme” file

for the availability of a specific version of Cloud SG software on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at https://www.tibco.com/patents.