Oracle® Tuxedo

Release Notes

Release 22c (22.1.1.0.0)
F70805-06
November 2024

Oracle Tuxedo Release Notes, Release 22c (22.1.1.0.0)

F70805-06

Primary Authors: Preeti Gandhe, Tulika Das

Contributors: Maggie Li

Contents

About the Oracle Tuxedo Release Notes

Oracle Tuxedo Release 22c lists the major new features and enhancements in the following topics.

In general, all content associated with Oracle Tuxedo Release 22c (22.1.0.0.0) applies to both Tuxedo 22.1.0.0.0 and Tuxedo 22.1.1.0.0, unless explicitly specified.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

1

What's New and Improved in Oracle Tuxedo 22.1.1.0.0

Oracle Tuxedo Release 22c (22.1.1.0.0) includes the following new major feature:

- Support for TLS 1.3: TLS is supported on all network connections within a Tuxedo domain.

> **See Also:**
>
> Quick Reference for TLS Support

2

What's New and Improved in Oracle Tuxedo (22.1.0.0.0)

Oracle Tuxedo Release 22c (22.1.0.0.0) includes the following new major features and enhancements:

- Security Enforcement: This release enhances the security features of Oracle Tuxedo.

  > **See Also:**
  >
  > Security Enforcement

- Easy integration with Oracle Database Application Continuity: This release the Tuxedo applications can leverage the Oracle Database Application Continuity feature more easily.

  > **See Also:**
  >
  > Tuxedo Application Leverage Oracle Database Application Continuity

- Secure use of SNMP: This release enhances SNMP security as it supports `AES` by default for privacy protocol.

  > **See Also:**
  >
  > Secure Use of SNMP

- The Tuxedo Server CLOPT -o and -e parameters support Tuxedo server IDs and Tuxedo process ID.

  > **See Also:**
  >
  > Tuxedo Server CLOPT -o and -e parameters support Tuxedo server and process IDs

- Other Updates in 22c

  > **See Also:**
  >
  > Other Updates in Oracle Tuxedo Release 22c (22.1.0.0.0)

- Security Enforcement
- Tuxedo Application Leverage Oracle Database Application Continuity

- Secure Use of SNMP
- Tuxedo Server CLOPT -o and -e parameters support Tuxedo server and process IDs
- Other Updates in Oracle Tuxedo Release 22c (22.1.0.0.0)

2.1 Security Enforcement

In this release, we introduce the following update of security capabilities to ensure that secure Tuxedo deployment by default.

- Mandatory Security Setting
- Link-Level Encryption
- Secure Sockets Layer
- JOLT Client
- Supported Algorithms for Public Key Security
- Default Use of TLS 1.2 with XAUTHSVR

> **See Also:**
>
> - Mandatory Security Setting
> - Link-Level Encryption
> - Secure Sockets Layer
> - JOLT Client
> - Supported Algorithms for Public Key Security
> - Default Use of TLS 1.2 with XAUTHSVR

2.1.1 Mandatory Security Setting

In Oracle Tuxedo Release 22c (22.1.0.0.0), the `SECURITY` parameter in the `UBBCONFIG` file is mandatory. If you set the value to NONE, a warning message appears in `ULOG`: `CMDTUX_CAT:8423: WARN:` Insecure option `NONE` is set for the `SECURITY` keyword. By setting `TM_SECURITY_CONFIG` to `NONE`, you indicate that the behavior in previous Tuxedo releases is desired: The `SECURITY` parameter is optional, and by default, it has the value `NONE`. No warning is reported to `ULOG` if the `SECURITY` value is `NONE`.

2.1.2 Link-Level Encryption

In this release, the LLE is disabled by default. Tuxedo client/server exits with an error, while detecting LLE in use instead of reporting a warning message in the User Log (ULOG). Setting the environment variable `TM_ALLOW_NOTLS` to `Y` allows you to enable LLE if you need it for some reason.

> **WARNING:**
>
> LLE is deprecated. Oracle recommends you to use SSL for securing your network links.

When using LLE, set the environment variable `LLE_DEPRECATION_WARN_LEVEL` to `NONE` or `ONCE` to suppress the warning message.

2.1.3 Secure Sockets Layer

The following components use TLS 1.2 at link level in the Oracle Tuxedo Release 22c (22.1.0.0.0) by default. The following components fail if SSL is unspecified as a command-line option:

- Set `CLOPT '-s'` to start the WSL.
- Set `CLOPT '-s'` to start the *JSL*.
- Set `CLOPT '-S'` to start the *ISL*.
- Set `CLOPT '-s'` to start the `tlisten`.

  `BRIDGE` fails to start if `OPTIONS` does not include the *SSL* setting in the `UBBCONFIG` file. `GWTDOMAIN` fails to start if `NWPROTOCOL` does not include the *SSL* or *SSL_ONE_WAY* setting in the `DMCONFIG`.

  By default, Tuxedo acts as an SSL client or server using TLS 1.2. To enable Tuxedo components to accept TLS 1.0 or 1.1 connections, use the environment variable `TM_TLS_FORCE_VER`.

  The Oracle Tuxedo Release 22c (22.1.0.0.0) supports the following cipher suites by default:

  - TLS_RSA_WITH_AES_256_CBC_SHA256
  - TLS_RSA_WITH_AES_256_GCM_SHA384
  - TLS_RSA_WITH_AES_128_CBC_SHA256
  - TLS_RSA_WITH_AES_128_GCM_SHA256

You can use `TM_CIPHERSUITES` environment variable to specify permitted cipher suites.

The minimum key length of the public key algorithm RSA is **2048** by default . Tuxedo detects the key length when loading the key/certificate, and fails the load if the key length is smaller than **2048**. To use a shorter key length, specify the minimum allowed key length in the environment variable `TM_MIN_PUB_KEY_LENGTH`.

`TM_ALLOW_NOTLS` can be set to Y to disable SSL/TLS connections for compatibility with the previous release. No encryption occurs at the link level if you set the min/max key length to (0,0).

2.1.4 JOLT Client

The JOLT Client uses the following to replace environment variables with Java properties.

- The Jolt client must connect to the Jolt server using TLS 1.2 by default. You can set Java Property `TM_ALLOW_NOTLS` to `Y` to allow the Jolt client to connect to a server that uses LLE or without encryption.

- You can use the `TM_MIN_PUB_KEY_LENGTH` Java property to specify the minimum allowed RSA key length. The default key length is **2048** if this property is not enabled.
- You can use the `bea.JOLT.tls.version` Java property to set a JOLT Client TLS versions. The default protocol version is **TLS1.2** if this property is not enabled.
- You can use the `bea.JOLT.tls.ciphersuites` Java property to specify Client cipher suites explicitly. You can set the `bea.JOLT.tls.ciphersuites` Java property to specify Client cipher suites explicitly. Please use the following cipher suites instead of the default:
  - TLS_RSA_WITH_AES_256_CBC_SHA256
  - TLS_RSA_WITH_AES_256_GCM_SHA384
  - TLS_RSA_WITH_AES_128_CBC_SHA256
  - TLS_RSA_WITH_AES_128_GCM_SHA256

2.1.5 Supported Algorithms for Public Key Security

Oracle Tuxedo Release 22c (22.1.0.0.0) supports the following Public Key Security algorithms:

- **Symmetric Key Algorithms**:
  1. Data Encryption Standard (DES)
  2. DES3
  3. RC2 (Rivest's Cipher 2)
  4. RC5
- **Asymmetric Key Algorithms**:
  1. Digital Signature Algorithm (DSA)
  2. Rivest, Shamir, and Adelman (RSA)
- **Message Digest Algorithms**:
  1. Message Digest (MD5)
  2. Secure Hash Algorithm 1 (SHA1)

> **Note:**
>
> Oracle Tuxedo Release 22c (22.1.0.0.0) includes a few insecure algorithms that are disabled by default. To enable backward compatibility, set the environment variable `TM_USE_OLD_CIPHER` to `Y` for backward compatibility reasons.

> **See Also:**
>
> Public Key Security

2.1.6 Default Use of TLS 1.2 with XAUTHSVR

In the Oracle Tuxedo Release 22c (22.1.0.0.0), XAUTHSVR uses SSL/TLS protocol to connect to LDAP servers. The default cipher-suites are set to `AES256-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES128-GCM-SHA256`. The default TLS version is set to `1.2`.

To modify the protocol, configure the `TLS_OPTIONS` within the OpenLDAP client using a configuration file or an environment variable. For more information, see OpenLDAP Configurations.

- Configure XAUTHSVR with WebLogic Server (WLS) 14.1.1

> **See Also:**
>
> Setting up the XAUTHSVR Server Configuration File

2.1.6.1 Configure XAUTHSVR with WebLogic Server (WLS) 14.1.1

`GAUTHSVR` is desupported in this release, an alternative is to use `XAUTHSVR`. Ensure that there is no existing `GAUTHSVR` configuration in the `UBBCONFIG` file, then follow the steps to configure `XAUTHSVR` with WLS (LDAP).

Setting Up the XAUTHSVR Server Configuration File

1. Open `UBBCONFIG` file with a text editor.
2. In `REASOURCES` section, perform the following:
   a. Set the `SECURITY` parameter to one of these values: `USER_AUTH` or `AUTHSVC`.
   b. Set the `OPTIONS` parameter to `EXT_AA`.
   c. Perform the following:
   - If the `SECURITY` parameter is set to `USER_AUTH`, set `AUTHSVC` to `AUTHSVC`, which is the service name advertised by the `XAUTHSVR` server.
3. Set up `A -f <fullpath-to-tpxauth>/tpxauth` in the `SERVERS` section.

   Following is the content of configuration in the `tpxauth` file:

```
FILE_VERSION 1
LDAP_VERSION 3
BINDDN cn=Admin
BASE ou=people,ou=myrealm,dc=mydomain     <- BaseDN for WLS embedded LDAP.
It might be changed according to WLS domain settings
LDAP_ADDR //<hostname or IP address>:<port>      <- WLS server listen
address
UID_KW uid
MEMBEROF_KW ismemberof
PWD_KW userPassWord
```

4. Encrypt the password in the `tpxauth` file:

```
tploadconf -f tpxauth
```

5. Enter password twice for the `WLS LDAP BINDDN`.

6. Set the environment variable `XAUTH_UID_DN_SUFFIX`

   - Set the `",<BASE>"` environment variable. Here, the `BASE` is the base DN defined in the `tpxauth`.

     ```
     export XAUTH_UID_DN_SUFFIX=",ou=people,ou=myrealm,dc=mydomain"
     ```

7. Set `tmloadcf` to `Y` to load the configuration. The `tmloadcf` command parses `UBBCONFIG` and loads the binary `TUXCONFIG` file to the location referenced by the `TUXCONFIG` variable and enter the password *twice* to access the application.
8. `tmboot` begins the Tuxedo application after passing the `ENCRYPTION_REQUIRED=Y` parameter.

2.2 Tuxedo Application Leverage Oracle Database Application Continuity

Application Continuity in Oracle Real Application Clusters (RAC), Oracle RAC One Node, and Oracle Active Data Guard hides outages from end users and applications by restoring the in-flight database sessions following recoverable outages. Application Continuity masks outages from end users and applications by recovering the in-flight work for impacted database sessions following outages. Application Continuity performs this recovery beneath the application so that the outage appears to the application as a slightly delayed execution. AC (Application Continuity) was introduced in Oracle DB 12.2. Starting with Oracle Database 19c, Transparent Application Continuity (TAC) transparently tracks and records session and transactional state so the database session can be recovered following recoverable outages. This is accomplished by not requiring application knowledge or application code changes, allowing Transparent Application Continuity to be enabled for your applications.

> **See Also:**
>
> Application Continuity

You have multiple ways to connect Oracle database in a Tuxedo server such as:

- XA connection

  You can invoke `tpopen()` parameter to create an XA connection to Oracle database.

- Oracle Call Interface (OCI) connection

  You can use OCI APIs for connecting to Oracle database.

- Oracle Pro*C connection

  You can use `EXEC SQL CONNECT` parameter for connecting to Oracle database.

Tuxedo applications utilize the AC feature with only OCI connection. Ensure that you have OCI 12.2. or higher version for AC support and similarly for TAC support.

How to use the AC feature

Follow the steps to configure to use the Application Continuity:

1. When AC is enabled on the Oracle Database side, and a Tuxedo server uses OCI APIs to connect to the Oracle Database explicitly, You can indicate whether or not to declare the

database request boundary to enable the application continuity feature. You can set the following parameter in the corresponding SERVERS section in Tuxedo UBBCONFIG:

```
ORAREQBOUNDARY = {Y | N}
```

The default is N.This attribute can also be specified in T_SERVER class through TM_MIB as shown in the following table:

Attribute	Type	Permissions	Values	Default
TA_ORAREQBOUNDARY	string	rw-r--r-	“{Y|N}”	"N"

2. When TAC is enabled at the Oracle Database side, and a Tuxedo server uses OCI APIs to connect to the Oracle Database explicitly, the Tuxedo server utilizes the AC feature no matter whether ORAREQBOUNDARY is configured or not, or to any value.

Benefits of Using the AC feature

When the Tuxedo application leverages Oracle Database AC, the Tuxedo server does not have to explicitly call OCI APIs to re-connect to the Oracle Database upon active node failure; instead of, DB connections re-initiate and automatically replay DB APIs, resulting in successful OCI calls.

> **Tip:**
>
> To leverage Tuxedo enhancements when interacting with Oracle Database using OCI APIs, ensure that you are following the steps:
>
> 1. Copy $TUXDIR/libs/tuxociucb.so.1.0 to *$ORACLE_HOME/lib/* and set the environment variable ORA_OCI_UCBPKG to: export ORA_OCI_UCBPKG=tuxociucb.
> 2. Enter the following to Tuxedo Server CLOPT in UBBCONFIG:
>
>    ```
>    -L libclntsh.so -F noECID
>    ```

2.3 Secure Use of SNMP

This release deprecates Oracle SNMP Agent Integrator. Oracle recommends you to not use it.

Oracle Tuxedo Release 22c (22.1.0.0.0) includes the following changes:

- SNMP v1 and SNMP v2 are disabled
- Default protocol for privacy protocol is changed to AES from DES.
  - Updates to arguments for snmpkey:
    - -x privProtocol

      This flag indicates the protocols for generated keys. Default protocol is AES 128-bit CFB mode. Valid values are:
      - AES: Indicates AES 128-bit CFB mode.
      - DES: Indicates CBC-DES.

- Updates to arguments for `snmpget`,`snmpgetnext`,`snmptest`,`snmptrap`,and `snmpwalk`:

  * `-x PrivProtocol`

    This flag sets the privacy protocol (`DES` or `AES`) used for encrypted SNMPv3 messages. The default `privProtocol` is `AES`.

> **See Also:**
>
> - Oracle SNMP Agent Integrator Commands
> - SNMP Information

2.4 Tuxedo Server CLOPT -o and -e parameters support Tuxedo server and process IDs

By using the Tuxedo Server `CLOPT -o` and `-e` parameters, you can redirect `stdout` and `stderr` to specific files.

`UBBCONFIG` Server `CLOPT -o` and `-e` parameters support the following placeholders when the environment `TM_STDOUTERR_EXT` is set to `Y`:

```
%SRVID%: Tuxedo server ID
%PROCID% : process ID
```

For example:

```
simpserv SRVGRP=GROUP1 SRVID=2341 MIN=2 MAX=2 CLOPT="-A -o mystdout.%SRVID% -
e mystderr.%PROCID%.log"
```

The `stdout` file names appear to be `mystdout.2341` and `mystdout.2342` respectively, and the `stderr` file names appear to be `mystderr.<pid>.log`.

2.5 Other Updates in Oracle Tuxedo Release 22c (22.1.0.0.0)

Oracle Tuxedo Release 22c (22.1.0.0.0) includes the following:

- The Tuxedo Java Server is now certified with the OpenJDK

3

Interoperability and Coexistence

The following list explains Oracle Tuxedo Release 22c (22.1.0.0.0) interoperability with previous releases:

- Oracle Tuxedo Release 22c (22.1.0.0.0) coexists in the same domain with Oracle Tuxedo 22c,12.x, 11.x, 10.x, and 9.x.
- Oracle Tuxedo Release 22c (22.1.0.0.0) supports interdomain interoperability with Oracle Tuxedo 22c,12.x, 11.x, 10.x, and 9.x.
- Oracle Tuxedo Release 22c (22.1.0.0.0) ATMI server connects with Workstation client from 22c, 12.x, 11.x, 10.x, 9.x,6.5
- Oracle Tuxedo Release 22c (22.1.0.0.0) Jolt server connects with Jolt client from 22c, 12.x, 11.x, 10.x, 9.x,6.5
- Oracle Tuxedo Release 22c (22.1.0.0.0) workstation client connects with Oracle Tuxedo ATMI servers running on Tuxedo 22c, 12.x, 11.x, 10.x, and 9.x
- Oracle Tuxedo Release 22c (22.1.0.0.0) Jolt client connects with Oracle Tuxedo Jolt server running on Oracle Tuxedo 22c, 12.x, 11.x, 10.x, and 9.x
- Oracle Tuxedo Release 22c (22.1.0.0.0) CORBA client connects with Oracle Tuxedo 22c, 12.x, 11.x, 10.x, and 9.x
- Oracle Tuxedo Release 22c (22.1.0.0.0) CORBA server connects with Oracle Tuxedo CORBA client running on Tuxedo 22c, 12.x, 11.x, 10.x, and 9.x

This is similar to Oracle Tuxedo 12.2.2, for information about the Oracle Tuxedo Interoperability, see Interoperability and Coexistence.

> **Note:**
>
> An Oracle Tuxedo clients cannot invoke each other.

4
Deprecated Features

The following is deprecated feature from the Oracle Tuxedo Release 22c (22.1.0.0.0):

Oracle SNMP Agent
Oracle recommends you to not use the Oracle SNMP agent in Oracle Tuxedo Release 22c (22.1.0.0.0).

> **See Also:**
>
> Oracle SNMP Agent Introduction

5
Desupported Features

The following features are desupported in Oracle Tuxedo Release 22c (22.1.1.0.0):

- Support for TLS 1.1 and older versions.

The following features are desupported in Oracle Tuxedo Release 22c (22.1.0.0.0):

GAUTHSVR
Starting from this release, Oracle desupports GAUTHSVR in favor of XAUTHSVR.

> **See Also:**
>
> Default Use of TLS 1.2 with XAUTHSVR

Service Component Architecture(SCA)
Starting from this release, Oracle desupports SCA.

> **See Also:**
>
> Oracle Tuxedo SCA Components

6
Upgrade Considerations

Oracle Tuxedo 12.2.2 applications (client and server) continue to run without re-compiling or re-linking, except for the following:

- Application that uses Xerces API
  - Problem statement: Xerces upgrade causes incompatibility at code level and binary level. Xerces-C++ 3.2.3 is an API-compatible, although not ABI-compatible, update to the 3.x branch. Code designed for use with Xerces 3 continues to compile, however, existing applications recompile to work with this version.
  - Action: Application that uses Xerces API needs to be recompiled with the new libtxml in Release Tuxedo 22c .

> **See Also:**
>
> * Migrating from Xerces-C++ 3.X to Xerces-C++ 3.2.3

7

Supported Platforms

Oracle Tuxedo software runs on the platforms listed in Oracle Tuxedo 22c Release Platform Data Sheets. Oracle has certified these platforms for development and production use with the Oracle Tuxedo Release 22c product. Oracle can provide customer support only for these platforms.

> **Note:**
>
> Although Oracle has attempted to implement the Oracle Tuxedo software in a manner that conforms to industry-standards, it is not feasible for Oracle to certify its use with all third-party databases, ORBs, and other products.

> **See Also:**
>
> Oracle Tuxedo 22c Release Platform Data Sheets.

8

Major Enhancements Post Oracle Tuxedo 12.2.2

The following sections describe major enhancements in Oracle Tuxedo Release 22c (22.1.0.0.0). Each enhancements include Bug number and its description and is listed by the BugDB number.

BugDB Number	Description
Bug 29132612	**Display `DOMAINID` in `WSH` and `tlisten` process command arguments.** WSH now displays `"-C dom=<DOMAINID>"` in process command arguments if `DOMAINID` is defined in `UBBCONFIG`. tlisten supports parameter `"-D dom=<DOMAINID>"` in command line options.
Bug 26198613	**`TMQUEUE_MQM` supports auto reconnection.** When `TMQUEUE_MQM` encounters the following errors, the current request will fail and the connection retry mechanism will be enabled (only if `MAXRETRIES` is not set to 0). When the next request arrives, `TMQUEUE_MQM` reconnects to the MQ. As a result, `TMQUEUE_MQM` will not retry at regular intervals, but only when it receives requests. The program will exit once `MAXRETRIES` has been reached.
Bug 23603910	**GWADM supports "-N" option.** The default `GWADM` behavior is: 1. If there is no `DMTLOG`, then create it. 2. If there is a damaged (uninitialized) `DMTLOG`, `GWADM` overwrites it with a good one. A newly-introduced `GWADM` option "`-N`" is available. If newly introduced is specified, `GWADM` behaviors are changed as follows: a. f there is no `DMTLOG`, `GWADM` fails at boot. b. If there is a damaged (uninitialized) `DMTLOG`, `GWADM` fails at the boot. `DMTLOG` is created by printing a message in ULOG.
Bug 23549348	Implement 64-bit XDR for `FML32 FLD_LONG`.
Bug 22857251	Support parallel startup by MIB.
Bug 22856094	**Workstation connection pool supports dynamic spawning of connections.** 1. `MAX_CONN_POOL` is a newly added environment variable for `WSCONNpoolinit`(int poolsize). If `MAX_CONN_POOL` is set before `WSCONNPOOLINIT()`, the maximum size of the connection pool is equal to `MAX_CONN_POOL` and the minimum size of the pool is set to poolsize. 2. Introduced a revised API, `WSCONNpoolinit2` (int minSize, int maxSize). The API creates a connection pool with a maximum size of maxSize and a minimum size of minSize. The pool creates a minimum number of connections if either of the above methods is used to set the maximum size. The maximum size of the pool will be reached when there are no more free connections in the pool.