Oracle® Tuxedo JCA Adapter

Users Guide

11*g* Release 1 (11.1.1.2)

November 2010

Oracle Tuxedo JCA Adapter Users Guide, 11*g* Release 1 (11.1.1.2)

Contents

Oracle Tuxedo JCA Adapter Users Guide

Oracle Tuxedo JCA Adapter Users Guide

This chapter contains the following topics:

Prerequisites

Installing and deploying the Oracle Tuxedo JCA Adapter requires the following prerequisites:

- Access to JDK 1.5 or later
- 5 MB free disk space
- A text editor or XML editor

Installing the Oracle Tuxedo JCA Adapter

The complete Oracle Tuxedo JCA Adapter is contained in a single resource archive file (`com.oracle.tuxedo.TuxedoAdapter.rar`) which you can download from the Oracle Web site. It contains both Java `.jar` files for resource adapter and standard JCA adapter deployment descriptor.

Table 1 lists the Oracle Tuxedo JCA Adapter resource archive file content.

Copy the downloaded Resource Archive file to a target directory with correct read and write access control. Un-jar the resource archive file to browse the contents and view the standard deployment descriptor.

Table 1 Resource Archive Content

Resource name	Description
`com.oracle.tuxedo.adapter_1.1.0.0.jar`	JAR file containing the Oracle Tuxedo JCA Adapter classes.

`com.bea.core.jatmi_1.3.1.0.jar`
Note: For WebLogic Server, `com.bea.core.jatmi_1.3.1.0.jar` must be exported using `EXT_PRE_CLASSPATH` to replace the one that come with WebLogic Server installation. This environmental variable must be set before starting WebLogic Server.
`export PATCH_CLASSPATH=$TUXEDO_JCA_ADAPTER_DIR/com.bea.core.jatmi_1.3.1.0.jar`
	JAR file containing the Java ATMI buffer types, and interfaces.

`com.bea.core.i18n_1.4.0.0.jar`	The I18N utility classes.
`javax.transaction_1.0.0.0_1-1.jar`	JTA 1.1 classes.
`javax.ejb_3.0.1.jar`	EJB 3.0 support classes.
`adapter.properties`	The Oracle Tuxedo JCA Adapter message catalogue.
`adapter_ja.properties`	The Oracle Tuxedo JCA Adapter Japanese Message Catalogue.

Table 1 Resource Archive Content

`tja.xsd`	The Oracle Tuxedo JCA Adapter configuration XML schema.
`dmconfig.xml`	The sample /Domain configuration.
`META-INF/ra.xml`	The sample Factory-Based configuration deployment descriptor for resource adapter.
`META-INF/client-side.ra.xml`	The sample client-side only deployment descriptor for resource adapter.
`META-INF/weblogic-ra.xml`	The sample WebLogic deployment descriptor for resource adapter.
`META-INF/sample.weblogic-ra.xml`	The sample WebLogic deployment descriptor with Factory-Based configuration for resource adapter.
`META-INF/MANIFEST.MF`	The Manifest file for Oracle Tuxedo JCA Adapter.
`META-INF/server.ra.xml.`	The `dmconfig`-based resource adapter deployment descriptor. You can choose to use `dmconfig` to configure the Oracle Tuxedo JCA Adapter and then modify the configuration using the sample `dmconfig.xml` and `server.ra.xml` file. **Note:** After modification, you must rename `server.ra.xml` to `ra.xml`.

Using the Oracle Tuxedo JCA Adapter

The Oracle Tuxedo Adapter can be configured using one of three ways. The first way is to use an XML-based configuration text file it is also called dmconfig. The second way is to use Resource Adapter Deployment Descriptor it is also called ra.xml. The third way is to use factory-based configuration, this type of configuration is different from application server vendor to application server vendor. User must choose one of the three ways to configure Oracle Tuxedo JCA Adapter. The mixed of different ways configuration is not supported.

The XML-based configuration file provides the way to configure using complete capabilities of Oracle Tuxedo JCA Adapter. It is suitable for users with complex configuration requirements. To configure using this method user must configure the `"resourceadapter-class"` with com.oracle.tuxedo.adapter.TuxedoResourceAdapter in the `"resourceadapter"` element of the Resource Adapter Deployment Descriptor.

Listing 1 shows a Resource Adapter Deployment Descriptor fragment that enables using an XML-based `dmconfiguration` file.

Listing 1 Resource Adapter Deployment Descriptor For Using XML-Based Configuration File

```
...
<resourceadapter>
  <resourceadapter-class>com.oracle.tuxedo.adapter.TuxedoResourceAdapter</
resourceadapter-class>
…
     <outbound-resourceadapter>
       <connection-definition>
<managedconnectionfactory-class>com.oracle.tuxedo.adapter.spi.TuxedoManage
dConnectionFactory</managedconnectionfactory-class>
…
   </outbound-resourceadapter>
</resourceadapter>
…
```

The Resource Adapter Deployment Descriptor, commonly known by its file name ra.xml, based configuration utilizes the custom properties in the deployment descriptor. It provides a subset of configuration capability of Oracle Tuxedo JCA Adapter that is suitable for client-side only operations. It provides an easiest way to configure an Oracle Tuxedo JCA Adapter.

To configure using Resource Adapter Deployment Descriptor method user must configure the `"resourceadapter-class"` with `com.oracle.tuxedo.adapter.TuxedoClientSideResourceAdapter` in the 'resourceadapter' element of the Resource Adapter Deployment Descriptor.

Listing 2 shows a Resource Adapter Deployment Descriptor fragment that enables using Custom Properties based configuration.

Listing 2 Resource Adapter Deployment Descriptor Using Custom Properties

```
<resourceadapter>

<resourceadapter-class>com.oracle.tuxedo.adapter.TuxedoClientSideResourceA
dapter</resourceadapter-class>
…
    <outbound-resourceadapter>
      <connection-definition>

<managedconnectionfactory-class>com.oracle.tuxedo.adapter.spi.TuxedoManage
dConnectionFactory</managedconnectionfactory-class>
…
  </outbound-resourceadapter>
</resourceadapter>
…
```

The factory-based configuration uses vendor specific way to configure connection factories. Each connection factory has its own configuration. It provides a larger subset of configuration capability of Oracle Tuxedo JCA Adapter when compares to Resource Adapter Deployment Descriptor based configuration. It provides an easier way to configure an Oracle Tuxedo JCA Adapter.

To configure using factory-based configuration method user must configure the `"resourceadapter-class"` with `com.oracle.tuxedo.adapter.TuxedoFBCResourceAdapter` in the `'resourceadapter'` element and the "`managedconnectionfactory-class`" in the 'connection-definition' of the '`outbound-resourceadapter`' element with value `com.oracle.tuxedo.adapter.spi.TuxedoFBCManagedConnectionFactory`.

Listing 3shows a Resource Adapter Deployment Descriptor fragment that enables using factory-based configuration.

Listing 3 Resource Adapter Deployment Descriptor Fragment

```
…
<resourceadapter>

<resourceadapter-class>com.oracle.tuxedo.adapter.TuxedoFBCResourceAdapter</resourceadapter-class>
…
  <outbound-resourceadapter>
    <connection-definition>

<managedconnectionfactory-class>com.oracle.tuxedo.adapter.spi.TuxedoFBCManagedConnectionFactory</managedconnectionfactory-class>
…
  </outbound-resourceadapter>
</resourceadapter>
```

If the /Domain configuration, dmconfig, is configured in the deployment descriptor file, but the `"resourceadapter-class"` class configured is not `com.oracle.tuxedo.adapter.TuxedoResourceAdapter` then the dmconfig information is to be ignored.

If the `resourceadapter-class` class configured is `TuxedoResourceAdapter` then all the resource adapter deployment-based configuration will be ignored.

This section contains the following topics:

- Default Configuration
- dmconfig Configuration
- Resource Adapter Deployment Descriptor-Based Configuration
- Factory-Based Configuration

Default Configuration

The "default" configuration is not applicable to every object types available whether it is the dmconfig configuration, Resource Adapter Deployment Descriptor configuration, or factory-based configuration. It is available for Local Access Point, Session Profile, Session, and Import. Resources, Remote Access Point and Export are not supported.

The "default" configuration works with `dmconfig` file-based configuration, resource adapter descriptor property based configuration, and factory based configuration. It makes it un-necessary to configure `LocalAccessPoint`, `SessionProfile`, `Session`, and import in some situations. In those instances user may only need to configure RemoteAccessPoint in the `dmconfig` file or `remoteAccessPontSpec` in the `Resource Adapter Deployment Descriptor` file, or `remoteAccessPointSpec` in the factory-based configuration file, this greatly enhances the usability of Oracle Tuxedo JCA Adapter.

This section contains the following topics:

- Default LocalAcessPoint
- Default SessionProfile
- Session

Default LocalAccessPoint

The default LocalAccessPoint allows user not to configure a LocalAccessPoint. It does not have a listening end point and it will not accept inbound connecting request from Oracle Tuxedo `GWTDOMAIN` gateway. The ConnectionPolicy can only be "`ON_STARTUP`".

In dmconfig based configuration and Deployment Descriptor based configuration there can have only one default LocalAccessPoint; however, for factory-based configuration each factory can have its own default LocalAccessPoint.

The default LocalAccessPoint when used will create a UUID-based `LocalAccessPointId`. This UUID-based `LocalAccessPointId` is written in a file with name `.lapid`, but for factory-based configuration then a file with name "`.lapid.<factory-name>`" will be created.

In dmconfig based configuration the default LocalAccessPoint does not support SSL. If user needs SSL then a LocalAccessPoint should be configured.

In Deployment Descriptor based configuration the default LocalAccessPoint support SSL in a limited fashion. User cannot configure Key Store, Identity Alias, and Trusted Certificate Store with password protection.

In factory-based configuration the default LocalAccessPoint supports SSL. User can specifies password for Key Store, Identity Alias, and Trusted Certificate Store.

Dynamic RemoteAccessPoint (RAP) Insertion

In order to make default LocalAccessPoint to work, Oracle Tuxedo GWTDOMAIN gateway configuration is required in order to make this simplified /Domain configuration to work.

GWTDOMAIN gateway must be modified to allow Dynamic RemoteAccessPoint (RAP) Registration. If DYNAMIC_RAP is set to YES, it will also update the in-memory database of the status of the connection from those dynamically registered RAP. If the connection from those dynamically registered RAP lost then the information about that RAP will be removed from the SHM database.

GWADM must be modified to process the DM MIB correctly to reflect the connection status of those dynamically registered RAP. When the connection from those dynamically registered RAP lost their entries in the SHM database will also be removed so that the DM MIB query can return the connection status correctly.

The dynamically registered RAP will be added to /DOMAIN configuration permanently. Their existence will only be known when the Session is established. Their existence will be lost when the connection is lost.

The DM_CONNECTION Oracle Tuxedo /Domain DMIB call returns all the connected dynamically registered RemoteAccessPoint. All other dynamically registered RemoteAccessPoint that are not connected will not be shown.

The OPENCONNECTION DMIB request will not be supported to connect to those dynamically registered RAP.

The CLOSECONNECTION Oracle Tuxedo /DMIB request closes the connection and remove the session from those dynamically registered RemoteAccessPoint, and returns its connection status as 'UNKNOWN.

The PERSISTENT_DISCONNECT type of CONNECTION_POLICY will be honored that means when PERSISTENT_DISCONNECT is in effect all connections request from any RAP, whether they are dynamically or non-dynamically registered, will be rejected.

Default SessionProfile

The adapter-wise default session profile is always created whether or not a SessionProfile is configured in the dmconfig configuration file. There is can only be one default SessionProfile for dmconfig based configuration, and Deployment Descriptor-based configuration, and it is the adapter-wise default SessionProfile.

The adapter-wise default SessionProfile can not be modified when using dmconfig based configuration; however, if user needs different SessionProfile other than the default one then user should configure the appropriate SessionProfile, and assign it to the target Session.

The adapter-wise default SessionProfile can be modified when using Deployment Descriptor based configuration using a set of custom properties to achieve it. Since there is no specific session profile can be configured explicitly when using Deployment Descriptor based configuration, this adapter-wise default SessionProfile will be used for all the Sessions.

The factory-based configuration adds the support for factory-wise default SessionProfile in addition to the adapter-wise default SessionProfile. User using this configuration method cannot modify the adapter-wise default SessionProfile; however, user is allowed to modify factory-wise default SessionProfile using a set of factory custom properties. If the default SessionProfile is not suitable for any connection factories created connection then user can configure the factory-wise default SessionProfile for each connection factories.

Table 2 lists the default configuration SessionProfile type elements.

Note: The SessionProfile related property configured in the resource adapter deployment descriptor file as shown in Listing 19 are used in the construction of the default SessionProfile.

Table 2 SessionProfile Type Elements

Element Name	Type	Default Value	Description
Security	string	NONE	The type of /Domain session authentication required.
BlockTime	int	60000	The maximum number of milliseconds allowed for a blocking outbound request using this profile.
Interoperate	boolean	false	Specifies whether the session is allowed to interoperate with a remote Tuxedo 6.5 release GWTDOMAIN gateway or not.
ConnectionPolicy	string	ON_STARTUP	The condition under which this GWTDOMAIN session is established.

Table 2 SessionProfile Type Elements

Element Name	Type	Default Value	Description
`CredentialPolicy`	`string`	`LOCAL`	The user credential propagation policy. When the value is `LOCAL`, then there is no propagation.
`RetryInterval`	`long`	`60`	The number of seconds that a session waits between automatic connection establishment attempts. This is meaningful only when `ConnectionPolicy` is set to `ON_STARTUP`.
`MaxRetries`	`long`	`9223372063857758078`	The maximum number of times that Oracle Tuxedo JCA Adapter tries to establish a session connection to remote Oracle Tuxedo access points. This is meaningful only when `ConnectionPolicy` is set to `ON_STARTUP`.
`CompressionLimit`	`int`	`2147483647`	The compression threshold a session uses when sending data to a remote Oracle Tuxedo access point. Application buffers larger than this size are compressed.
`MinEncryptBits`	`string`	`0`	The minimum encryption key length (in bits) a session uses after establishing a session connection. A value of 0 indicates no encryption is needed. Key strength of 256 bits is for SSL support only.
`MaxEncryptBits`	`string`	`128`	The maximum encryption key length (in bits) a session uses to transport data after establishing a session connection. A value of 0 indicates no encryption is needed. Key strength of 256 bits is for SSL support only.

Table 2 SessionProfile Type Elements

Element Name	Type	Default Value	Description
KeepAlive	long	0	Specifies whether a GWTDOMAIN session is configured with application level keep alive, and its maximum idle time before wait timer start ticking. Measured in milliseconds.
KeepAliveWait	long	10000	Specifies whether a session requires the application level keep alive acknowledgement, and how long it will wait without receiving acknowledgement before declaring the connection inaccessible

Default Session

If resource adapter deployment descriptor based configuration or factory-based configuration is used, or there is no Session configured in the dmconfig file then the session will be implicitly created between all local access points and all the remote access points. This is called default Session. If default Session is used then it will and can only use adapter-wise default `SessionProfile` when Deployment Descriptor based or dmconfig based configuration is used. The factory-based configuration will use factory-wise default `SessionProfile`, any factory not configured with its own default `SessionProfile` then the adapter-wise default `SessionProfile` will be used.

For instance, if there are 2 RemoteAccessPoint configured and the default LocalAccessPoint is used and there is no Session configured then two default Session are created.

Default Import

Oracle Tuxedo JCA Adapter allows you to access remote Oracle Tuxedo services or resources through the configured sessions even when there is no Oracle Tuxedo service or resource configured. This feature is called Default Import. If there is at least one configured Oracle Tuxedo service or resource, then this feature is automatically disabled and only the request targets configured Oracle Tuxedo services or resources can be forwarded to Oracle Tuxedo.

When there is no Oracle Tuxedo service or resource configured as Import in the Oracle Tuxedo JCA Adapter configuration, then request filtering is not performed and all requests are forwarded to Oracle Tuxedo using the name specified in the service request invocation. If more than one session is configured or created implicitly by Oracle Tuxedo JCA Adapter, then all the service requests are load-balanced among those configured sessions using a RoundRobin algorithm.

All Oracle Tuxedo services/resources must be accessible through the Oracle Tuxedo JCA Adapter (not only they are available in all Oracle Tuxedo application domains, but also accessible through all configured sessions)

dmconfig Configuration

The Oracle Tuxedo JCA Adapter configuration file is an XML based file represented by a property with name 'dmconfig' in the resource adapter deployment descriptor (`ra.xml`). This property value can be either an absolute path to a configuration file or it can be represent as a resource of the resource `archive.rar` file. User needs to use this method for configuration when full blown client and server operations are required.

Listing 4 shows an example that tells the Oracle Tuxedo JCA Adapter to use a 'dmconfig' configuration file. The full path name to the configuration file is: `/home/work/adapter/dmconfig.xml`.

Listing 4 dmconfig Full Path Example

```
...
<resourceadapter>
  <resourceadapter-class>com.oracle.tuxedo.adapter.TuxedoResourceAdapter</
   resourceadapter-class>
  <config-property>
     <config-property-name>dmconfig</config-property-name>
     <config-property-type>java.lang.String</config-property-type>
     <config-property-value>/home/work/adapter/dmconfig.xml</config-proper
       ty-value>
  <config-property>
...
```

Listing 5 shows an example telling the Oracle Tuxedo JCA Adapter that it uses a 'dmconfig' configuration file and it is packaged as part of the resource archive with the resource file name 'dmconfig.xml'. However, if this configuration resource file is not found in the archive it will be treated as a configuration file located in the current working directory.

Listing 5 dmconfig Archive

```
...
<resourceadapter>
   <resourceadapter-class>com.oracle.tuxedo.adapter.TuxedoResourceAdapter<
    /resourceadapter-class>
<config-property>
   <config-property-name>dmconfig</config-property-name>
   <config-property-type>java.lang.String</config-property-type>
   <config-property-value>dmconfig.xml</config-property-value>
<config-property>
...
```

TuxedoConnector Root Element

There is only one root element, `"TuxedoConnector"`, in the Oracle Tuxedo JCA Adapter configuration file; it is represented by the complex type `TuxedoConnectorType`. It contains the following elements as listed in Table 3:

- Resources
- Local Access Point
- Remote Access Point
- Session Profile
- Session
- Import
- Export

Table 3 TuxedoConnectorType Element

Element Name	Type	Occurrence	Description
`Resources`	`ResourceType`	`0..1`	Oracle Tuxedo JCA Adapter environment
`LocalAccessPoint`	`LocalAccessPointType`	`0..unbounded`	Local access point
`RemoteAccessPoint`	`RemoteAccessPointType`	`0..unbounded`	Remote access point
`SessionProfile`	`SessionProfileType`	`0..unbounded`	Session profile including QoS configuration information.
`Session`	`SessionType`	`0..unbounded`	Defines the possible session/connection to a remote domain.
`Import`	`ImportType`	`0..unbounded`	Defines a remote resource available to the adapter
`Export`	`ExportType`	`0..unbounded`	Defines a local resource available to remote domain.

The configuration must use the `<TuxedoConnector>` and `</TuxedoConnector>` tags as shown in Listing 6.

Listing 6 TuxedoConnector Tags Example

```
<?xml version="1.0" encoding="UTF-8"?>
<TuxedoConnector>
...
</TuxedoConnector>
```

Resources

The Resources element is represented by ResourceType. It specifies the resource adapter execution environment.Only one Resource element can be configured in the Oracle Tuxedo JCA Adapter configuration file. Table 4 lists the "ResourceType" elements.

Table 4 ResourceType Element

Element Name	Type	Occurrence	Description
FieldTable16Classes	string	0..unbounded	Fully qualified field table 16 classes for FML.
FieldTable32Classes	string	0..unbounded	Fully qualified field table 32 classes for FML32.
ViewFile16Classes	string	0..unbounded	Fully qualified VIEW table 16 classes for VIEW.
ViewFile32Classes	string	0..unbounded	Fully qualified VIEW table 32 classes for VIEW32.
ApplicationPasswordEncrypted	string	0..1	The application password for joining an Oracle Tuxedo application. The password is encrypted. The password length cannot exceed 30 characters and should be the same as Oracle Tuxedo.
TpusrFile	string	0..1	The TPUSR file full path name.
RemoteMBEncoding	string	0..1	The TPUSR file full path name.
MBEncodingMapFile	string	0..1	Full encoding map file path name.

Note: There is no attribute defined for any "ResourceType" element; however, the "TPUsrFile" element can be overridden by the TpusrFile element in the RemoteAccessPoint.

Listing 7 shows a "Resources" configuration example.

Listing 7 Resources Configuration Example

```
<Resources>
  <FieldTable16Classes>tuxedo.test.fml16.FieldTbl16</FieldTable16Classes>
  <ViewFile32Classes>tuxedo.test.simpapp.View32</ViewFile32Classes>
  <ApplicationPasswordEncrypted>tuxpassword</ApplicationPasswordEncrypted>
</Resources>
```

If "ApplicationPassowrdEncrypted" is configured, it is required to run the provided utility `com.oracle.tuxedo.tools.DMConfigChecker` to encrypt the password, and optionally generate a key store. In this case, the `keyFileName` must point to a valid key store file or key store resource. If keyFileName is not configured in the Resource Adapter Deployment Descriptor then Oracle Tuxedo JCA Adapter fails to decrypt the password.

Listing 8 shows an example that tells Oracle Tuxedo JCA Adapter that a key store resource is configured.

Listing 8 Key Store Resource Example

```
...
<resourceadapter>
  <resourceadapter-class>com.oracle.tuxedo.adapter.TuxedoResourceAdapter</
   resourceadapter-class>
  <config-property>
     <config-property-name>dmconfig</config-property-name>
     <config-property-type>java.lang.String</config-property-type>
     <config-property-value>dmconfig.xml</config-property-value>
  <config-property>
  <config-property>
     <config-property-name>keyFileName</config-property-name>
     <config-property-type>java.lang.String</config-property-type>
     <config-property-value>foo.key</config-property-value>
  <config-property>
...
```

Local Access Point

The Local Access Point element is represented by `LocalAccessPointType`. It specifies a listening address and possible link-level failover address. Table 5 lists the `LocalAccessPointType` elements.

Table 5 LocalAccessPointType Element

Element Name	Type	Occurrence	Description
`AccessPointId`	`string`	`0..1`	The connection principal name. It must be *globally* unique. **Note:** Globally unique means the identification specified in the configuration must be unique within all the interconnected Oracle Tuxedo Domains and Application Servers through Oracle Tuxedo GWTDOMAIN gateway, WebLogic Server WTC, and Oracle Tuxedo JCA Adapter. If not specified, it uses the *locally* unique `LocalAccessPoint “name”` attribute value (see Listing 9).
`NetworkAddress`	`string`	`1..unbounded`	The local access point listening address including both host address and port number. Specify the TCP/IP address in the format `//hostname:port or //#.#.#.#:port`.
`SSLInfo`	`complex Type`	`0..unbounded`	Specifies whether SSL encryption is used to ensure communication or not. If not configured, LLE is used.

The `SSLInfo` element is an anonymous complex type. If it is included in the configuration, all its elements must be configured and SSL is used as the transport mechanism. If it is not included, TCP/IP is used as the transport mechanism and uses link-level encryption for data privacy if encryption is required. Table 6 lists `SSLInfo` elements.

Table 6 SSLInfo Elements

Element Name	Type	Occurrence	Description
`MutualAuthenticationRequired`	`boolean`	`0..1`	Specifies if client authentication is required for SSL communication. The default value is `"false"`.
`IdentityKeyStoreFileName`	`string`	`1..1`	Full identity key store file path name.
`IdentityKeyStorePassPhraseEncrypted`	`string`	`1..1`	Password used to encrypt the identity key store.
`PrivateKeyAlias`	`string`	`1..1`	Alias in the identity key store used to retrieve private key.
`PrivateKeyPassPhraseEncrypted`	`string`	`1..1`	Encrypted password used to decrypt the private key in the identity key store.
`TrustKeyStoreFileName`	`string`	`1..1`	Full trust key store file path name.
`TrustKeyStorePassPhraseEncrypted`	`string`	`1..1`	Passphrase used to decrypt the trust key store when retrieving certificates.

The `“name”` attribute is defined for `LocalAccessPointType`. It is used to identify the configuration record as represented by `LocalAccessPointType`. It specifies a *locally* unique local access point name as shown in Listing 9.

Listing 9 LocalAccessPoint Name Attribute

```
<LocalAccessPoint name="LDOM1">
  <LocalAccessPointId>Godfried</LocalAccessPointId>
 <NetworkAddress>//neocortex:14001</NetworkAddress>
</LocalAccessPoint>
```

'LDOM1' must be locally unique. 'Gotfried' must be globally unique. If the 'AccessPointId' element is not specified, the value of the name attribute is used as 'AccessPointId'. In this scenario, the value of the name attribute must be globally unique.

By default SSL only authenticates the server (connection request responder). However, to enable the client authentication the "MutualAuthenticationRequired" element must be set to "true."

Remote Access Point

The Remote Access Point element is represented by RemoteAccessPointType. It defines a network address for remote Oracle Tuxedo Domain access points. Table 7 lists RemoteAccessPointType elements.

Table 7 RemoteAccessPointType Element

Element Name	Type	Occurrence	Description
AccessPointId	string	0..1	It is used by default as connection principal name that is used to identify this remote access point when attempting to establish a session with remote Tuxedo access point. It must be *globally* unique. **Note:** Globally unique means the identification specified in the configuration must be unique within all the interconnected Oracle Tuxedo Domains and Application Servers through Oracle Tuxedo GWTDOMAIN gateway, WebLogic Server WTC, and Oracle Tuxedo JCA Adapter If not specified, it uses the *locally* unique RemoteAccessPoint "name" attribute value (see Listing 10).
NetworkAddress	string	1..unbounded	The host network address and port number of the remote Oracle Tuxedo access point. Specify the TCP/IP address in the format //hostname:port or //#.#.#.#:port.
TpusrFile	string	0..1	The full path name to a TPUSR file for this remote access point.

Table 7 RemoteAccessPointType Element

Element Name	Type	Occurrence	Description
AllowAnonymous	boolean	0..1 Value set {true, false}	Indicates whether the remote Tuxedo access point allow anonymous access or not. The default value is "false".
DefaultApplicationKey	string	0..1	The default application key value for the remote Oracle Tuxedo access point. If not specified, then "-1" is assumed.
CustomApplicationKey	string	0..1	Configures the custom application key generator. If not specified, the default application key generator is used.

The CustomApplicationKey element is an anonymous complex type. If not specified, the default Application Key plug-in is used. If specified, all of its elements must be configured and the Custom Application Key plug-in class is loaded for every session that communicates with the remote access point. Table 8 lists CustomApplicationKey elements.

Table 8 CustomApplicationKey Element

Element Name	Type	Occurrence	Description
ApplicationKeyClass	string	1..1	The fully qualified application key generator class name.
ApplicationKeyClassParam	string	1..1	The parameter string passed to the application key generator when it is initialized at runtime

The “name” attribute is defined for RemoteAccessPoint. It specifies the *locally* unique Remote Access Point Name. Listing 10 shows a RemoteAccessPointType name attribute example.

Listing 10 RemoteAccessPoint Name Attribute Example

```
<RemoteAccessPoint name="RDOM1">
  <AccessPointId>Geneve</AccessPointId>
  <NetworkAddress>//bluestar:11023</NetworkAddress>
  <TpusrFile>/tja/lady-geneve/tpusr</TpusrFile>
</RemoteAccessPoint>
```

'RDOM1' must be locally unique. 'Geneve' must be globally unique. If the 'AccessPointId' element is not specified, the value of the name attribute is used as 'AccessPointId'. In this scenario, the value of the name attribute must be globally unique.

Session Profile

The Session Profile element is represented by SessionProfileType. It contains all the QoS parameters for a TDOMAIN session between an Oracle Tuxedo JCA Adapter Local Access Point and an Oracle Tuxedo Remote Access Point. Table 9 lists SessionProfileType elements.

Table 9 SessionProfileType Element

Element Name	Type	Occurrence	Description
Security	string	0..1 Value set {NONE, APP_PW, DM_PW}	The type of /Domain authentication required. The default value is NONE.
BlockTime	int	0..1 Value range 0..2147483647	The maximum number of milliseconds allowed for a blocking outbound request using this profile. The default value is 60000.
Interoperate	boolean	0..1 Value set {true, false}	Specifies whether the session is allowed to interoperate with a remote Tuxedo 6.5 release GWTDOMAIN gateway or not. The default value is "false".
ConnectionPolicy	string	0..1 Value set {ON_DEMAND, ON_STARTUP, INCOMING_ONLY}	The condition under which this GWTDOMAIN session is established. The default value is ON_DEMAND.
ACLPolicy	string	0..1 Value set {LOCAL, GLOBAL}	The ACL policy to be enforced on this GWTDOMAIN session. The default value is LOCAL.

Table 9 SessionProfileType Element

Element Name	Type	Occurrence	Description
CredentialPolicy	string	0..1 Value set {LOCAL, GLOBAL}	The user credential propagation policy. When the value is Local, then there is no propagation. The default value is LOCAL.
RetryInterval	long	0..1 Value range 0..2147483647	The number of seconds that sessions wait between automatic connection establishment attempts. Set this element value only when ConnectionPolicy is set to ON_STARTUP.
MaxRetries	long	0..1 Value range 0..922372063857758	The maximum number of times that a session tries to establish a session connection to remote Oracle Tuxedo access point. Set this element only when ConnectionPolicy is set to ON_STARTUP. Default value is 922337206385775807.
CompressionLimit	int	0..1 Value range 0..2147483647	The compression threshold a session uses when sending data to a remote Oracle Tuxedo Access Point. Application buffer larger than this size are compressed. Default value is 2147483647.
MinEncryptBit	string	0..1 Value set {"0", "40", "56", "128", "256"}	The minimum encryption key length (in bits) a session uses when establishing a session connection. A value of 0 indicates no encryption is accepted. Key strength of 256 bits is for SSL support only. Default value is 0.
MaxEncryptBit	string	0..1 Value set {"0", "40", "56", "128", "256"}	The maximum encryption key length (in bits) a session uses when establishing a session connection. A value of 0 indicates no encryption is accepted. Key strength of 256 bits is for SSL support only. Default value is 128.

Table 9 SessionProfileType Element

Element Name	Type	Occurrence	Description
KeepAlive	long	0..1 Value range {0..2147483647}	Specifies if a GWTDOMAIN session is configured with Application Level Keep Alive, and its maximum idle time before wait timer start ticking. Measured in milliseconds. Default value is 0.
KeepAliveWait	long	0..1 Value range {0..2147483647}	Specifies if a session requires the Application Level Keep Alive acknowledgement, and how long it will wait without receiving acknowledgement before declaring the connection inaccessible. Measured in milliseconds.Default value is 10000.

The “name” attribute is defined for SessionProfileType. It is used by the Session object to get the correct session profile. Listing 11 shows a SessionProfileType name attribute example.

Listing 11 SessionProfile Name Attribute Example

```
<SessionProfile name="profile1">
  <Security>DM_PW</Security>
  <ConnectionPolicy>ON_STARTUP</ConnectionPolicy>
  <ACLPolicy>Global</ACLPolicy>
  <CredentialPolicy>Global</CredentialPolicy>
  <RetryInterval>100</RetryInterval>
</SessionProfile>
```

Session

The Session element is represented by "SessionType". It specifies a permissible connection between a Local Access Point and a Remote Access Point. Only one session can be configured between a Local Access Point and a Remote Access Point. Table 10 lists the SessionType elements.

Table 10 SessionType Elements

Element Name	Type	Occurrence	Description
LocalAccessPointName	string	1..1	The local access point that is used to compose a TDOMAIN session. This "LocalAccessPoint" refers to the "name" attribute of a “LocalAccessPoint” element.
RemoteAccessPointName	string	1..1	The remote access point that is used to compose a TDOMAIN session. This "LocalAccessPoint" refers to the "name" attribute of a “LocalAccessPoint” element.
ProfileName	string	0..1	The profile to be used for a session. If not specified uses adapter-wise default session profile.
PasswordPair	complex Type	0..2	The password pair used when SECURITY equals DM_PW to authenticate the session.

The PasswordPair element is an anonymous complex type. At most, two password pairs can be configured. It allows user to configure passwords for Tuxedo Domain Session Authentication. Table 11 lists the PasswordPair elements.

Table 11 PasswordPair Element

Element Name	Type	Occurrence	Description
LocalPasswordEncrypted	string	1..1	The encrypted local password. The password length cannot exceed 30 characters.
RemotePasswordEncrypted	string	1..1	The encrypted remote password. The password length cannot exceed 30 characters.
ActivationTime	string	0..1	The date and time string used to indicate when a password pair becomes effective. If not specified, then it is assumed already become effective. The format is YYYY:MM:DD:hh:mm:ss.
DeactivationTime	string	0..1	The date and time string used to indicate when a password pair becomes obsolete. If not specified, it is assumed that it will never expire. Same format as ActivationTime.

The “name” attribute is defined for SessionType. It is used to identify a TDOMAIN session. Listing 12 shows a SessionType name attribute example.

Listing 12 SessionType Name Example

```
<Session name="session1_1">
  <LocalAccessPointName>LDOM1</LocalAccessPointName>
  <RemoteAccessPointName>RDOM1</RemoteAccessPointName>
  <ProfileName>profile1</ProfileName>
</Session>
```

If no Session is configured, by default, the Oracle Tuxedo JCA Adapter creates sessions between all Local Access Points and all Remote Access Points. These dynamically created sessions can only use the default Session Profile.

Import

The Import element is represented by ImportType. It identifies an existing remote Tuxedo Application Domain resource that can be accessed by the Oracle Tuxedo JCA Adapter client. Table 12 lists ImportType elements.

Table 12 ImportType Element

Element Name	Type	Occurrence	Description
RemoteName	string	0..1	The actual remote Tuxedo resource name being exported by the Oracle Tuxedo TDomain gateway. If not specified, it has the same value as the "name" attribute.
SessionName	string	1..unbounded	The name of the session that imports a resource from the remote Oracle Tuxedo application domain.
LoadBalancing	string	0..1 Value set {RoundRobin, Random}	The load balancing algorithm used for an imported resource. The default value is RoundRobin.

There are three defined attributes: name, autotran, and trantime.

- name: specifies the resource name to used by a TPCALL or function name for "execute()" CCI interface.
- autotran: enables/disables of AUTOTRAN. It only accepts true or false values. If it is set to true then when this imported resource is to be invoked by Oracle Tuxedo JCA Adapter client outside a global or local transaction then the Oracle Tuxedo JCA Adapter starts a transaction with a remote Oracle Tuxedo Domain.

  If it is set to false, the Oracle Tuxedo JCA will not start a transaction if the client service request is outside of global or local transaction; however, by setting it to false does not prevent client service request of this remote resource to be a participant of a global or local transaction.

  If it is not configured then the adapter-wise AUTOTRAN property in the resource adapter deployment descriptor will be used to determine the AUTOTRAN.
- trantime: specifies the transaction timeout value for AUTOTRAN of this resource. The value is measured in seconds. If not specified and AUTOTRAN is required then the 'appManagedLocalTxTimeout' property of the resource adapter deployment descriptor will be used.

  If appManagedLocalTxTimeout is not specified then JVM property com.oracle.tuxedo.adapter.appManagedLocalTxTimeout is used.

  If com.oracle.tuxedo.adapter.appManagedLocalTxTime JVM property is not specified, it defaults to 300 seconds.

  Listing 13 shows an ImportType name example that describes an imported resource with name TUXUPPER. The AUTOTRAN transaction timeout is set to 10 seconds

Listing 13 ImportType Name Example

```
<Import name="TUXUPPER" autotran="true" trantime="10">
   <RemoteName>TOUPPER_1</RemoteName>
   <SessionName>session1</SessionName>
   <LocalBalancing>RoundRobin</LoadBalancing>
</Import>
```

Export

The Export element is represented by ExportType. It specifies a local resource that is accessible from a remote Oracle Tuxedo Application Domain. Table 13 lists ExportType elements.

Table 13 ExportType Element

Element Name	Type	Occurrence	Description
RemoteName	string	0..1	Resource name the Oracle Tuxedo application uses to access service in an application server. If not specified, it has the same value as the name attribute.
SessionName	string	1..unbounded	The session that allows access to the local resource.
Type	string	0..1 Value set {EJB, POJO}	Type of resource. The default is EJB.
Source	string	1..1	This is the EJB name or the target class of the POJO. It must be specified.
SourceLocation	string	0..1	This is the target jar for POJO. Its value is ignored if service type is EJB.

The "name" attribute is attribute is defined for ExportType. It is used to identify an exported resource. Listing 14 shows an ExportType name example.

Listing 14 ExportType Name Example

```
<Export name="tolower">
   <SessionName>session1</SessionName>
   <RemoteName>wtolower</RemoteName>
   <Type>EJB</Type>
   <Source>TolowerEJB</source>
</Export>
```

Resource Adapter Deployment Descriptor-Based Configuration

The main component used to deploy and repackage the resource archive for deployment is the resource adaptor deployment descriptor (the `ra.xml` file in the `META-INF` directory). The resource adapter deployment descriptor must be configured before repacking the resources into a resource archive.

This XML-based text file and can be modified by using a text or XML editor. The downloaded Oracle Tuxedo JCA Adapter contains a simple version of the deployment descriptor to assist you configuring the Oracle Tuxedo JCA Adapter deployment.

For Deployment Descriptor based configuration user must specify an Oracle Tuxedo JCA Adapter deployment configuration through using the deployment descriptor file (ra.xml). There is a set of custom properties to allow user to specify the configuration. All properties specified in this way their scope will be adapter-wise.

Resource Adapter Deployment Descriptor Properties

The deployment descriptor-based configuration style is based on standard simple property types: `config-property-name`, `config-property-type`, and `config-property-value`. These property types cannot be repeated. They are available in the Resource Adapter Deployment Descriptor "`resourceadapter`" file.

The resource adapter deployment descriptor configuration method supports the following types of properties:

- Resource-Related Properties
- Local Access Point-Related Properties
- RemoteAccessPoint-Related Properties
- SessionProfile-Related Properties
- Import-Related Property

Resource-Related Properties

Most of the Resources elements specified in the `dmconfig` file are available for deployment descriptor-based configuration.

Table 14 lists the resource adapter deployment descriptor `Resources` properties.

Table 14 Resources Properties

Property	Type	Default Value	Description
fieldTable16Classes	String	None	Field table 16 classes for FML. It is a comma-separated list.
fieldTable32Classes	String	None	Field table 32 classes for FML32. It is a comma-separated list.
viewFile16Classes	String	None	VIEW Table 16 classes for VIEW. It is a comma-separated list.
viewFile32Classes	String	None	VIEW Table 32 classes for VIEW32. It is a comma-separated list.
tpusrFile	String	None	The full path name to the TPUSR file.
remoteMBEncoding	String	None	The Multi-Byte encoding used by Oracle Tuxedo.
mBEncodingMapFile	String	None	Full path name to the encoding map file.

Listing 15 shows a configuration example that describes two VIEW32 classes information using resource adapter deployment descriptor- based custom property configuration.

Note: This causes the Oracle Tuxedo JCA Adapter to use the `viewFile32Classes` with full package name accessible through `SYSTEM CLASSPATH`.

Listing 15 ra.xml File VIEW32 Custom Property Configuration

```
…
<config-property>
  <config-property-name>viewFile32Classes<config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>tuxedo.view32.view1,tuxedo.view32.view2
   </config-property-value>
```

```
</config-property>
…
```

Local Access Point-Related Properties

The majority of the `LocalAccessPoint` type elements are not available in a resource adapter descriptor-based configuration; however, a single `localAccessPointSpec` can be specified.

Table 15 lists the resource adapter deployment descriptor `LocalAccessPoint` properties.

Table 15 LocalAccessPoint Properties

Property	Type	Default Value	Description
`localAccessPointSpec`	`String`	`None`	The syntax is `//<network address>:<port>/domainId=<domain id>` The domain id is used by default as connection principal name. It, and it has to be globally unique.
`identityKeyStoreFileName`	`String`	`None`	The full path name of the identity key store file name.
`privateKeyAlias`	`String`	`None`	The alias in the identity key store to be used to retrieve private key.
`trustedKeyStoreFileName`	`String`	`None`	The full path name of the trusted key store file name.

The `localAccessPointSpec` property is optional. When specified in a non-clustered environment then it is useful if you want to have all the configuration information in the resource adapter deployment descriptor file. However, when it is specified in a clustered environment where the configuration is copied to all cluster nodes, all Oracle Tuxedo JCA Adapters have the same access point identification.In this situation, the connection behavior becomes unpredictable and is not supported.

To get around this clustered environment problem, do not to configure it at all then the `default LocalAccessPoint` is created with UUID-based `LocalAccessPointId`. This UUID-based `LocalAccessPointId` is written in a file with name `.lapid` in the current working directory.

Listing 16 provides an `localAccessPointSpec` custom configuration example.

Listing 16 LocalAccessPoint Custom Property Configuration Example

```
<config-property>
  <config-property-name>localAccessPointSpec</config-property-name>
  <config-property-type>java.lang.String<config-property-type>
  <config-property-value>//localhost:12345/domainId=jdom_id
  </config-property-value>
</config-property>
```

This example specifies a `LocalAccessPoint` with `AccessPointName` and `AccessPointId` equal to `jdom_id` is created and listens for the incoming connection requests at port 12345 on the local host.

If the `default LocalAccessPoint` is created, the connection policy can only be `"ON_STARTUP"`, the listening endpoint is not created.

When "`default LocalAccessPoint`" is constructed by the Oracle Tuxedo JCA Adapter dynamically there is no listening endpoint and it will be purely for Client-Side only operation mode. There is no incoming connection request possible.

SSL/TLS can also be supported through the resource adapter deployment descriptor using the 3 properties related to identity and key stores. However, there are two limitations; firstly, it will only support identity and trusted key stores without password protection; secondly, all the Oracle Tuxedo JCA Adapter instances in a cluster will have the same values. If this is not desirable user must either modify the resource adapter deployment descriptor for all the nodes that requires different behavior or change to use `dmconfig` method to configure the clustered Oracle Tuxedo JCA Adapter.

RemoteAccessPoint-Related Properties

The `RemoteAccessPoint` is represented by a text string that contains both networking address and per `RemoteAccessPoint` access control-related information; however, in resource adapter deployment descriptor property based configuration it can be represented by a single remoteAccessPointSpec property. The remoteAccessPointSpec property will be a comma-separated list and a comma separates each remote access point. Each remote access point is represented in a specific format described in the sample after the following table.

Table 16 lists the resource adapter deployment descriptor RemoteAccessPoint-related properties.

Table 16 RemoteAccessPoint Properties

Property	Type	Default Value	Description
remoteAccessPointSpec	String	None	This contains both NetworkAddress and AccessPointId plus the name attribute. This is a comma-separated list of RemoteAccessPoint. The domainId in each entry is used to replace AccessPointId. This domainId is used by default as connection principal name that is used to identify a remote access point when attempting to establish a session with remote Tuxedo access point. If not specified then it is an error in Deployment Descriptor based configuration; and the name has to be globally unique.
rapAllowAnonymous	Boolean	{true, false}	Indicates whether the remote Tuxedo access point allow anonymous access or not. The default value is "false".
rapDefaultApplicationKey	String	Any valid Tuxedo application key.	The default application key value for the Oracle Tuxedo access point. If not specified then "-1" is assumed.

Table 16 RemoteAccessPoint Properties

Property	Type	Default Value	Description
`rapApplicationKeyClass`	`String`	`None`	The fully qualified class name of the custom application key generator. If not specified then the default application key generator will be used.
`rapApplicationKeyClassParam`	`String`	`None`	The parameter string passed to the custom application key generator when the class is initialized at runtime.

If property `rapApplicationKeyClass` is not specified then `rapApplicationKeyClassParam` will be ignored if one is configured.

To support link level failover parenthesis is used to group failover addresses together for a `RemoteAccessPoint`. The first address specified in a group will be the primary address. The second address in a group backs up the first address, and the third address in a group backs up the second address and so on.

Listing 17 provides an example that configures two `RemoteAccessPoint`. The first is accessible through domain `guinevre` with network address `//bluestar:11023`. The second is accessible through domain `galahad` with network address `//orion:37456`.

Listing 17 RemoteAccessPoint Custom Property Configuration Example

```
<config-property>
  <cofig-property-name>remoteAccessPointSpec</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>//bluestar:11023/domainId=guinevre,//orion:37456/
   domainId=galahad</config-property-value>
</config-property>
```

Both remote domains `guinevre` and `galahad` will have the same QoS associate with them. In this case if `rapApplicationKeyClass` and `rapApplicationKeyClassParam` are specified they will be treated as if they are available to both RemoteAccessPoint guinevre and galahad, user must make the rapApplicationKeyClass available to both RAP with same fully qualified class path otherwise the Oracle Tuxedo JCA Adapter will fail to start.

Listing 18 provides an example that configures two RemoteAccessPoint with failover address. The first is accessible through domain guinevre with primary network address //bluestar:11023, and the back up network address //orion:12345.

The second is accessible through domain galahad with network address //orion:37456, and the back up network address //bluestar:37456.

Listing 18 RemoteAccessPoint Custom Property Configuration Example

```
<config-property>
  <config-property-name>remoteAccessPointSpec</config-porperty-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>(//bluestar:11023,//orion:12345)/domainId=guinevr
   e,(//orion:37456,//bluestar:37456)/domainId=galahad
   </config-property-value>
</config-property>
```

There can be one and only one remoteAccessPointSpec property be specified in the resource adapter deployment descriptor file. If there are more than one configured then the application server's JCA container will only honor the last one configured.

There is no default RemoteAccessPoint so if remoteAccessPointSpec property is not configured then there is no dynamically created RemoteAccessPoint and this will make Oracle Tuxedo JCA Adapter useless even though it still can be started. To configure RemoteAccessPoint through remoteAccessPointSpec property the "resourceadapter-class" element in the resource adapter deployment descriptor file must be configured using TuxedoClientSideResourceAdapter.

SessionProfile-Related Properties

The information contained in Session Profile can be represented by one set of configuration properties. It will contain all the QoS parameters for a TDOMAIN session between an Oracle Tuxedo JCA Adapter Local Access Point and an Oracle Tuxedo Remote Access Point.

Table 17 lists the resource adapter deployment descriptor SessionProfile properties.

Table 17 SessionProfile Properties

Property	Type	Default Value	Description
spBlockTime	Integer	0..2147483647	The maximum number milliseconds allowed for a blocking outbound request using this profile. The default value is 60000 milliseconds. This can be overridden by the IntercationSpec setExecutionTime() parameter.
spInteroperate	Boolean	{true, false}	Specify whether this session is allowed to interoperate with a remote Tuxedo 6.5 release GWTDOMAIN gateway or not. The default value is "false".
spCredentialPolicy	String	{LOCAL, GLOBAL}	The user credential propagation policy. When set to "LOCAL", there is no propagation. Default value is "LOCAL".
spRetryInterval	Long	0..2147483647	The number of seconds that the session waits between automatic connection establishment attempts. Default value is 60.
spMaxRetries	Long	0..922337206385775807	The maximum number of times that this sessions tries to establish a session connection to remote Tuxedo access point. Default value is 922337206385775807
spCompressionLimit	Integer	0..2147483647	The compression threshold all the sessions use when sending user data to a remote Tuxedo access point. Application buffers larger than this size are compressed. The default value is 21474483647.
spMinEncryptBits	String	{"0","40","56","128","256"}	The minimum encryption key length in bits all the session use after establishing a session connection. A value of 0 indicates encryption may not required. The default value is "0".

Table 17 SessionProfile Properties

Property	Type	Default Value	Description
spMaxEncryptBits	String	{"0","40","56","128","256"}	The maximum encryption key length in bits all the sessions use after establishing a session connection. A value of 0 indicates no encryption is required. The default value is "128"
spKeepAlive	Long	{0..2147483647}	Specifies whether all the sessions are configured with Application Level Keep Alive, and its maximum idle time before wait timer start ticking. Default value is "0", it means disable Application Level Keep Alive. The measurement is in millisecond. When connection is busy there is no need to send special keep alive message to remote gateway; however, when there is spKeepAlive number of milliseconds, rounded up to second, without activities over the connection the a special keep alive message will be sent and a timer of spKeepAliveWait, also rounded up to second, will be started. If no acknowledgement received during this "wait" time the connection is declared dead, and connection will be closed, session will be terminated.
spKeepAliveWait	Long	{0..2147483647}	This tells whether this session requires the acknowledgement of Application Level Keep Alive or not, and how long it will wait without receiving acknowledgement before declaring the connection is inaccessible. The default value is 10 seconds. The measurement is in milliseconds. A value of 0 will disable wait timer.

Listing 19 provides a `SessionProfile` configuration example.

Listing 19 SessionProfile Custom Properties Configuration Example

```
< config-property>
  <config-property-name>spBlockTime</config-property-name>
  <config-property-type>java.lang.Integer</config-property-type>
  <config-property-value>120000</config-property-value>
</config-property>
```

All these `SessionProfile` related property configured in the resource adapter deployment descriptor file is used in the construction of the default `SessionProfile`.

Import-Related Property

The resource adapter deployment descriptor file based configuration will utilize the Default Import; however, it also provide ability to restrict what can be access from adapter to remote Tuxedo application domain in a uniformly way. A single `"impResourceName"` property can be specified and it contains a comma-separated list of remote Tuxedo services/resources allowed to access.   One restriction applies is that these will be applied to all the sessions that are possible.

The `LoadBalancing` algorithm cannot be specified and uses `RoundRobin`. If `impResourceName` is specified then there is no default Import be created by Oracle Tuxedo JCA Adapter.

Table 18 lists the resource adapter deployment descriptor Import-Related property.

Table 18 Import-Related Property

Property	Type	Default Value	Description
`impResourceName`	`String`	`Any valid Tuxedo resource name in a comma-separated list.`	The valid Tuxedo service name or queue name. There will be no name transaction done for the wire protocol. This is basically the same as the `RemoteName` attribute of an Import in the `dmconfig` file.

Listing 20 provides an `impResourceName` example. This property is limited to the resource adapter deployment descriptor based configuration so a "resourceadapter-class" must be configured with `TuxedoClientSideResourceAdapter`.

Listing 20 impResourceName Example

```
<config-property>
 <config-property-name>impResourceName</config-property-name>
 <config-property-type>java.lang.String</config-property-name>
 <config-property-value>TOUPPER_1,ECHO</config-property-value>
</config-property>
```

Session

There is no Session related property defined in the Resource Adapter Deployment Descriptor custom property configuration method. If a resource adapter deployment descriptor-based configuration is used or there is no Session configured in the dmconfig file, a session is implicitly created between all local access points and all the remote access points. This is called a “default Session.” If a default Session is used, it can only use the default `SessionProfile`.

For example, if there are two RemoteAccessPoint elements configured and the default LocalAccessPoint is used and there is no Session configured, then two default Sessions are created.

Export

There is no Export related property defined in the Resource Adapter Deployment Descriptor custom property configuration method.There is no default Export that supports inbound request from an Oracle Tuxedo application domain to application. This is true for resource adapter deployment descriptor custom properties-based configuration, factory-based configuration, dmconfig configuration file-based configuration.

Factory-Based Configuration

The factory-based configuration is similar to Resource Adapter Deployment Descriptor based configuration that they both utilize custom property to configure an Oracle Tuxedo JCA Adapter. The factory-based configuration provides a better configuration support using a larger set of custom properties.

The factory-based configuration consists of two major parts. The first part is adapter-wise properties that need to be configured in the Deployment Descriptor file in the "resourceadapter" using custom property. The second part is factory-wise properties that can be configured using different ways with different Java application server. For WebSphere they are configured through custom properties page of the "J2C connection factory", for WebLogic they are configured in weblogic-ra.xml.

Normally, user configures the Oracle Tuxedo JCA Adapter after it has been installed. User can configure Resource Adapter Deployment Descriptor, and then configure custom factory properties.

Properties In Resource Adapter Deployment Descriptor

There is a set of custom properties in the ResourceAdapter Deployment Descriptor are supported by factory-base configuration. They are there because they are all adapter-wise properties.

These properties is based on standard simple property types: config-property-name, config-property-type, and config-property-value. These property name cannot be repeated. They are available in the Resource Adapter Deployment Descriptor's "resourceadapter".

The resource adapter deployment descriptor custom properties for factory-based configuration supports the following types of properties:

Adapter-Wise Properties

Resource-Related Properties

Adapter-Wise Properties

Adapter-wise properties are available to all the connection factory.   There are few adapter-wise properties available to factory-based configuration; however, they can be overridden by property of the same name in the factory configuration. If these properties are not configured in a factory then factory will use these adapter-wise properties configuration.

Table 19 lists these adapter-wise custom properties in the resource adapter deployment descriptor that are supported by factory-based configuration.

Table 19 Adapter-Wise Properties

Property	Type	Default Value	Description
`autoTran`	`Boolean`	`false`	Defines whether AUTOTRAN is allowed or not. It will be used by factory if factory-wise autoTran is not configured.
`appManagedLocalTxTimeout`	`Integer`	`300 seconds`	Define the transaction timeout used by AUTOTRAN or client application managed local transaction. It will be used by factory if factory-wise `appManagedLocalTxTimeout` is not configured.
`throwFailureReplyException`	`Boolean`	`true`	Configure whether a `TuxedoFailureReplyException` will be thrown or not if a failure reply is received from Tuxedo. It will be used by factory if factory-wise `throwFailureReplyException` is not configured.

The following is the precedence order for AUTOTRAN transaction timeout in factory-based configuration.

1. factory-wise `appManagedLocalTxTimeout` property
2. adapter-wise `appManagedLocalTxTimeout` property
3. `com.oracle.tuxedo.adapter.AppManagedLocalTxTimeout` JVM property
4. default to 300 seconds

The adapter-wise "`appManagedLocalTxTimeout`" is configured in "resourceadapter" type in the Resource Adapter Deployment Descriptor(RADD), a.k.a. ra.xml, file as "config-property".

Listing 21 shows an example using AUTOTRAN with transaction timeout in ra.xml file.

Listing 21 using AUTOTRAN with Transaction Timeout in ra.xml File Example

```
<resourceadapter>
```

```
<resourceadapter-class>com.oracle.tuxedo.adapter.TuxedoFBCResourceAdapter<
/resourceadapter-class>
  <config-property>
    <config-property-name>autoTran</config-property-name>
    <config-property-type>java.lang.Boolean</config-property-type>
    <config-property-value>true</config-property-value>
  </config-property>
  <config-property>
    <config-property-name>appManagedLocalTxTimeout</config-property-name>
    <config-property-type>java.lang.Integer</config-property-type>
    <config-property-value>50</config-property-value>
  <config-property>
…
```

Resource-Related Properties

The Resources related properties are available to every factories and configured using Resource Adapter Deployment Descriptor. They are configured in the "resourceadapter" type in the Deployment Descriptor. The only exception is Application Password that it is made available to each factory for flexibility and it is not available here in Resources-Related properties.

All properties listed in Resource-Related Properties Table 20 are optional.

Table 20 Resource-Related Properties

Property	Type	Default Value	Description
fieldTable16Classes	String	None	Field table 16 classes for FML. It is a comma-separated list.
fieldTable32Classes	String	None	Field table 32 classes for FML32. It is a comma-separated list.

Table 20 Resource-Related Properties

Property	Type	Default Value	Description
viewFile16Classes	String	None	VIEW table 16 classes for VIEW. It is a comma-separated list.
viewFile32Classes	String	None	VIEW table 32 classes for VIEW32. It is a comma-separated list.
tpusrFile	String	None	The full path name to the TPUSR file.
remoteMBEncoding	String	None	The Multi-Bytes encoding used by Tuxedo Application.
mBEncodingMapFile	String	None	Full path name to the encoding map file.

Listing 22 shows a configuration example that describes two VIEW32 classes information in the Resource Adapter Deployment Descriptor for factory-based configuration.

Listing 22 Resource Adapter Deployment Descriptor Factory-based Configuration

```
…
<resourceadapter>

<resourceadapter-class>com.oracle.tuxedo.adapter.TuxedoFBCResourceAdapter<
/resourceadapter-class>
  <config-property>
   <config-property-name>viewFile32Classes<config-property-name>
   <config-property-type>java.lang.String</config-property-type>
   <config-property-value>tuxedo.view32.view1,tuxedo.view32.view2
</config-property-value>
  </config-property>
…
```

Properties For Connection Factory

There are three types of property can be configured for a connection factory. The first type is factory property also available as adapter-wise. The second type is property unique to factory-based configuration. The third type is regular configuration property.

- Adapter-Wise Properties Also Available In Factory
- Connection Factory Name
- Application Password
- Local Access Point Related Properties
- RemoteAccessPoint Related Properties
- RemoteAccessPoint Related Properties
- SessionProfile Related Properties
- Import Related Property
- Session
- Export

Adapter-Wise Properties Also Available In Factory

A few of adapter-wise properties can be specified for each factory configuration, these properties will override the adapter-wise properties. If they are not specified in a factory, the factory will use the adapter-wise configuration for these properties.

Table 21 Factory-Wise Property Table

Property Name	Type	Default Value	Description
autoTran	Boolean	None	Defines whether AUTOTRAN is available for requests using this factory. If not configured then use adapter-wise AUTOTRAN configuration. If configured, whether it is true or false, will override adapter-wise AUTOTRAN configuration.
appManagedLocalTxTimeout	Integer	None	Define the transaction timeout used by AUTOTRAN or client application managed transaction. If not configured then use adapter-wise local TX timeout. It is measured in second.
throwFailureReplyException	Boolean	None	Configure whether a TuxedoFailureReplyException will be thrown or not if a failure reply received from Tuxedo. If it is not configured then adapter-wise setting will be used.

The following is the precedence order for AUTOTRAN in factory-based configuration.

1. factory appManagedLocalTxTimeout property
2. adapter-wise appManagedLocalTxTimeout property
3. com.oracle.tuxedo.adapter.AppManagedLocalTxTimeout JVM property
4. default to 300 seconds

The following is the example using WebLogic's weblogic-ra.xml file.

Listing 23 Adapter-Wise weblogic_ra-xml File

```
…
<outbound-resource-adapter>
  <connection-definition-group>
```

```
<connection-factory-interface>javax.resources.cci.ConnectionFactory</conne
ction-factory-interface>
    <connection-instance>
      <jndi-name>eis/TuxedoConnectionFactory1</jndi-name>
        <connection-properties>
          <properties>
            <property>
              <name>autoTran</name>
              <value>true</value>
            </property>
            <property>
              <name>appManagedLocalTxTimeout</name>
              <value>50</value>
            </property>
…
```

Connection Factory Name

A connection factory name can be specified by using `connectionFactoryName` property. Although this property is optional, it is recommended to configure it if transaction is possible for service request originated from this connection factory, and also if user want to use DMMIB to configure a `DM_REMOTE_DOMAINS` in Tuxedo /Domain configuration dynamically.

Table 22 Connection Factory Property Table

Property Name	Type	Default Value	Description
`connectionFactoryName`	`String`	`None`	Defines the name of this connection factory.

The following is the example using WebLogic's weblogic-ra.xml file.

Listing 24 Connection Factory weblogic_ra-xml File

```
…
<outbound-resource-adapter>
  <connection-definition-group>

<connection-factory-interface>javax.resources.cci.ConnectionFactory</conne
ction-factory-interface>
    <connection-instance>
      <jndi-name>eis/TuxedoConnectionFactory1</jndi-name>
        <connection-properties>
          <properties>
            <property>
              <name>autoTran</name>
              <value>true</value>
            </property>
           <property>
             <name>connectionFactoryName</name>
            <value>TuxedoConnectionFactory1</value>
          </property>
…
```

If this property is configured and default LocalAccessPoint is configured then a file with the name "`.lapid.<connectionFactoryName>`" will be created in the current working directory which will contains the LocalAccessPoint Id generated dynamically. For instance using above example a file with the name "`.lapid.TuxedoConnectionFactory1`" will be created.

Application Password

The application password of the Resources, which is not supported in RADD based configuration, is added to factory-based configuration as factory-wise property. This property is available per factory, this is to facilitate the ability of different factory to have different Application Password. There is no equivalent in RADD based configuration.

The applicationPassword property for factory-based configuration can be in either clear text or cipher text. To configure it using cipher text user must use the output of `com.oracle.tuxedo.tools.EncryptPassword`. The following is the sample output:

```
c:\tuxedo\JCA\adapter> java -classpath %classpath%
com.oracle.tuxedo.tools.EncryptPassword mypassword foo.key Encrypted
Password: {Salted-AES}WBGk6LjHuI515pwXPTfaOQ==
```

For WebSphere 7.0 there is no need to use this tool to encrypt password since WebSphere will encrypt it for user.

Table 23 Resource Property Available in Factory

Property Name	Type	Default Value	Description
`applicationPassword`	`String`	`None`	Tuxedo Application Password in either clear text, or cipher text using com.oracle.tuxedo.tools.EncryptPassword tool.

Local Access Point Related Properties

Table 24 is the table for LocalAccessPoint related properties for the factory-based configuration.

Table 24 Properties of LocalAccessPoint

Property Name	Type	Default Value	Description
`localAccessPointSpec`	`String`	`None`	It is used by default as connection principal name. It has to be globally unique. `//<network address>:<port>/domainId`
`mutualAuthenticationRequired`	`Boolean`	`false`	Indicate whether mutual authentication is required when connecting to remote Tuxedo GWTDOMAIN gateway. This for SSL.
`identityKeyStoreFileName`	`String`	`None`	The full path name of the identity key store file name. This is for SSL.

Table 24 Properties of LocalAccessPoint

Property Name	Type	Default Value	Description
`identifyKeyStorePassPhrase`	`String`	`None`	The password for the identity key store in either clear text or cipher text. This is for SSL.
`privateKeyAlias`	`String`	`None`	The alias in the identity key store to be used to retrieve private key. This is for SSL.
`privateKeyPassPhrase`	`String`	`None`	The password, in either clear text or cipher text, used for decrypt the private key in the identity key store. This is for SSL.
`trustKeyStoreFileName`	`String`	`None`	The full path name of the trusted key store file name. This is for SSL.
`trustKeyStorePassPhrase`	`String`	`None`	The password, in either clear text or cipher text, to be used when retrieving certificate from trust key store. This is for SSL.

There are 7 SSL related properties, and 6 of them that are related to Key/Certificate store must be configured if SSL is required; the only one that is optional for using SSL is `mutalAuthenticationRequired` By default the `"mutualAuthenticationRequire"` is false. If anyone of the 6 required properties is missing then SSL will be ignored and thus no SSL, and depends on session profile information plus session negotiation with remote Tuxedo GWTDOMAIN gateway the LLE may be used.

It is optional to specify `localAccessPointSpec` property. If it is not specified then default LocalAccessPoint will be used. When default LocalAccessPoint is used for this factory it is recommended to also configure `connectionFactoryName`.

Listing 25 shows the example of using WebLogic's `weblogic-ra.xml` file.

Listing 25 weblogic-ra.xml Usage Example

```
…
<outbound-resource-adapter>
  <connection-definition-group>
```

```
<connection-factory-interface>javax.resources.cci.ConnectionFactory</conne
ction-factory-interface>
    <connection-instance>
      <jndi-name>eis/TuxedoConnectionFactory1</jndi-name>=
        <connection-properties>
          <properties>
            <property>
              <name>localAccessPointSpec</name>
              <value>//localhost:123456/domainId=JDOM</value>
            </property>
…
```

RemoteAccessPoint Related Properties

The RemoteAccessPoint is represented by both networking address and per RemoteAccessPoint Access Control related information. The most important property is `remoteAccessPointSpec`, it is a comma-separated list, and a comma separates each RemoteAccessPoint. Each RemoteAccessPoint is represented in a specific format described in the sample after the following table. In order to make the factory usable a `"remoteAccessPointSpec"` must be configured.

The following are the RemoteAccessPoint related properties that are available in factory-based configuration.

Table 25 RemoteAccessPoint Property Table

Property Name	Type	Value Range	Description
remoteAccessPointSpec	String	No default value.	This property contains both NetworkAddress and AccessPointId plus the name attribute. This is a comma separated list of RemoteAccessPoint. The domainId in each entry is used to replace AccessPointId. This domainId is used by default as connection principal name that is used to identify a remote access point when attempting to establish a session with remote Tuxedo access point. If not specified then it is an error in Client-Side only operation mode, and it has to be globally unique. The NetworkAddress also become part of the specification. The NetworkAddress contains both host network address and port number of the remote Tuxedo access point. Specify the TCP/.IP address in the format of //hostname:port or //#.#.#.#:port.
rapAllowAnonymous	Boolean	{true, false}	Indicates whether the remote Tuxedo access point allow anonymous access or not. The default value is "false".
rapDefaultApplicationKey	String	Any valid Tuxedo application key.	The default application key value for this remote Tuxedo access point. If not specified then "-1" is assumed.
rapApplicationKeyClass	String	No default value.	The fully qualified class name of the custom application key generator. If not specified then the default application key generator will be used.
rapApplicationKeyClassParam	String	No default value.	The parameter string passed to the custom the application key generator when the class is initialized at runtime.

The following is the example using the WebLogic weblogic-ra.xml file.

Listing 26 RemoteAccessPoint weblogic-ra.xml File

```
…
<outbound-resource-adapter>
  <connection-definition-group>

<connection-factory-interface>javax.resources.cci.ConnectionFactory</conne
ction-factory-interface>
    <connection-instance>
      <jndi-name>eis/TuxedoConnectionFactory1</jndi-name>
        <connection-properties>
          <properties>
            <property>
              <name>autoTran</name>
              <value>true</value>
            </property>
            <property>
              <name>localAccessPointSpec</name>
              <value>//localhost:123456/domainId=JDOM</value>
            </property>
           <property>
            <name>remoteAccessPointSpec</name>

<value>//bluestar:11023/domainId=guinevre,//orion:37654/domainId=galahad<v
alue>
          </property>
…
```

There can be one and only one `remoteAccessPointSpec` be specified for each factory. If `rapApplicationKeyClass` and `rapApplicationKeyClassParam` are specified they will be used for identity propagation for both guinevre and galahad.

The remoteAccessPointSpec

The `remoteAccessPointSpec` property has been enhanced to be able to configure more RemoteAccessPoint and Session related attributes. Comma is used to separate these attributes. Each attribute is a name and value pair. The following is the list of attributes supported.

- `domainId` - The remote access point Id. It must be specified.
- `lPasswd1` - Local password of password pair 1, can be in either clear text or cipher text.
- `lPasswd2` - Local password of password pair 2, can be in either clear text or cipher text.
- `rPasswd1` - Remote password of password pair 1, can be in either clear text or cipher text.
- `rPasswd2` - Remote password of password pair 2, can be in either clear text or cipher text.

At least one password pair must be valid if session authentication is "DM_PW". If both password pair 1 and password pair 2 are valid then password pair one will be used to encrypt session authentication information.

The lPasswd1, lPasswd2, rPasswd1, and rPasswd2 attributes for factory-based configuration can be in either clear text or cipher text. To configure it using cipher text user must use the output of com.oracle.tuxedo.tools.EncryptPassword. The following is the sample output:

```
c:\tuxedo\JCA\adapter> java -classpath %classpath%
com.oracle.tuxedo.tools.EncryptPassword mypassword foo.key Encrypted
Password: {Salted-AES}WBGk6LjHuI515pwXPTfaOQ==
```

The following is the example using WebLogic's weblogic-ra.xml file.

Listing 27 remoteAccessPointSpec weblogic-ra.xml File

```
…

<outbound-resource-adapter>

  <connection-definition-group>


<connection-factory-interface>javax.resources.cci.ConnectionFactory</conne
ction-factory-interface>
```

```
    <connection-instance>
      <jndi-name>eis/TuxedoConnectionFactory</jndi-name>
      <connection-properties>
        <properties>
          <property>
            <name>spSecurity</name>
            <value>DM_PW</name>
          </property>
          <property>
            <name>remoteAccessPointSpec</name>

<value>//localhost:123456/domainId=TUX_ID,lPasswd1=weblogic,rPassword=tuxe
do</value>
            </property>
…
```

SessionProfile Related Properties

The following is the table with default value of the default `SessionProfile` for factory-based configuration.

Table 26 SessionProfile Property Table

Property Name	Type	Value Range	Description
spBlockTime	Integer	0..2147483647	The maximum number of milliseconds allowed for a blocking outbound request using this profile. The default value is 60 seconds, which is 60000 milliseconds. The InteractionSpec set execution time method can override this block timeout.
spSecurity	String	{APP_PW,DM_PW,NONE}	If not configured then GWTDOMAIN session authentication is not required. When DM_PW session authentication security type is configured then at least one password pair must be configured for every remoteAccessPoint. If APP_PW session authentication security type is configured then user must configure applicationPassword for the factory.
spInteroperate	Boolean	{true, false}	Specify whether this session is allowed to interoperate with a remote Tuxedo 6.5 release GWTDOMAIN gateway or not. The default value is "false".
Connection Policy	Not available		Will always behave like ON_STARTUP.
spCredentialPolicy	String	{LOCAL. GLOBAL}	The user credential propagation policy. When it's value Is "LOCAL" then no propagation. Default value is "LOCAL".
spRetryInterval	Long	0..2147483647	The number of seconds that this session waits between automatic connection establishment attempt. Set this element value only when ConnectionPolicy is "ON_STARTUP". Default value is 60. The value 0 disables the connection retry mechanism.

Table 26 SessionProfile Property Table

Property Name	Type	Value Range	Description
`spMaxRetries`	`Long`	`0..922337206385775807`	The maximum number of times that this session tries to establish a session connection to remote Tuxedo access points. Set this element only when `ConnectionPolicy` is `"ON_STARTUP"`. Default value is 922337206385775807.
`spCompressionLimit`	`Integer`	`0..2147483647`	The compression threshold this session uses when sending data to a remote Tuxedo Access Point. Application buffers larger than this size are compressed. The default value is 2147483647.
`spMinEncryptBits`	`String`	`{"0","40","56","128","256"}`	The minimum encryption key length (in bits) this session uses when establishing a session connection. A value of 0 indicates no encryption is needed. Default value is "0".
`spMaxEncryptBits`	`String`	`{"0","40","56","128","256"}`	The maximum encryption key length (in bits) this sessions uses when establish a session connection. A value of 0 indicates no encryption is used. The default value is "128".

Table 26 SessionProfile Property Table

Property Name	Type	Value Range	Description
`spKeepAlive`	`Long`	`{0..2147483647}`	Specifies whether this GWTDOMAIN session is configured with Application Level Keep Alive, and its maximum idle time before wait timer start ticking. Default value is "0", and it means application level keep alive is disabled. The measurement is in millisecond. When connection is busy there is no need to send special keep alive message to remote gateway; however, when there is `spKeepAlive` number of milliseconds, rounded up to second, without activities over the connection then a special keep alive message will be sent and a timer of spKeepAliveWait (rounded up to second) will be started, if no acknowledgement received during this "wait" time the connection is declared dead, connection will be closed, and session will be terminated.
`spKeepAliveWait`	`Long`	`{0..2147483647}`	This tells whether this session requires the acknowledgement of Application Level Keep Alive or not, and how long it will wait without receiving acknowledgement before declare the connection is inaccessible. Default value is 10 seconds. If the value specified is '0' then there will be no checking of the acknowledgement from RemoteAccessPoint; this can prevent the session connection be closed by KeepAlive feature. The measurement is in millisecond.

The following is the WebLogic example of weblogic-ra.xml:

Listing 28 SessionProfile Property welogic-ra.xml File

```
…
<connection-instance>
```

```
…
  <connection-properties>
…
    <properties>
      <property>
        <name>spBlockTime</name>
        <value>120000</value>
      </property>
…
</connection-instance>
```

Import Related Property

Factory-based configuration can utilize the default Import; however, it will also provide ability to restrict what can be accessed from adapter to remote Tuxedo Application Domain in a uniformly way. A single `"impResourceName"` property can be specified for each factory and it contains a comma-separated list of remote Tuxedo service/resources the Oracle Tuxedo JCA Adapter client is allowed to access.

The LoadBalancing algorithm cannot be specified and it will always use RoundRobin. User's service requests will be load balanced among all the `RemoteAccessPoints` of that particular connection factory.

The following is the new property related to Import in the factory-based configuration.

Table 27 Import Related Property weblogic-ra.xml File

Property Name	Type	Value Range	Description
`impResourceName`	`String`	`Any valid Tuxedo resource` name `in a comma-separated list.`	The valid Tuxedo service name or queue name. There will be no name translation done for the wire protocol. This is essential the same as the `RemoteName` attribute of an Import in the dmconfig file.

Listing 29 shows the WebLogic example of weblogic-ra.xml:

Listing 29 Weblogic-ra.xml Usage Example

```
…
<connection-instance>
…
  <connection-properties>
    <properties>
      <property>
        <name>impResourceName</name>
        <value>TOUPPER,ECHO</value>
      </property>
…
</connection-instance>
```

Session

A session connects an Oracle Tuxedo JCA Adapter LocalAccessPoint to a remote Tuxedo `GWTDOMAIN` gateway. In factory-based configuration there is no need to specify session explicitly so there is no "Session" related property available, and all the sessions available in a factory will be default Session. The Oracle Tuxedo JCA Adapter will create a session for every possible LocalAccessPoint and RemoteAccessPoint combinations for a connection factory. Since there can only be one LocalAccessPoint per connection factory configuration so there can be at most 1xN number of sessions possible where the 'N' is the number of RemoteAccessPoint. For instance, if your configured the following, using WebLogic's weblogic-ra.xml as example

Listing 30

```
<property>
  <name>localAccessPointSpec</name>
  <value> //localhost:12345/domainId=JDOM</value>
</property>
<property>
  <name>remoteAccessPointSpec</name>

<value>//localhost:13456/domainId=TDOM1,//blues:23457/domainId=TDOM2</valu
e>
</property>
```

Then the following sessions are possible.

```
(JDOM,TDOM1)
(JDOM,TDOM2)
```

Export

There is no "Export" related property available for factory-based configuration.

How To Configure Factory-Based Configuration

WebLogic Server

The Tuxedo JCA Adapter factory-based configuration is done through WebLogic's weblogic-ra.xml file. The sample fragments of the configuration can be found in previous sections;Listing 31 lists a complete sample configuration.

Listing 31 Complete Sample Configuration

```
<?xml version="1.0"?>
<weblogic-connector
 xmlns="http://www.bea.com/ns/weblogic/90">
  <jndi-name>eis/TuxedoConnector</jndi-name>
  <enable-access-outside-app>true</enable-access-outside-app>
  <enable-global-access-to-classes>true</enable-global-access-to-classes>
```

```
      <outbound-resource-adapter>
        <connection-definition-group>

    <connection-factory-interface>javax.resource.cci.ConnectionFactory</connection-factory-interface>
          <connection-instance>
            <jndi-name>eis/TuxedoConnectionFactory1</jndi-name>
            <connection-properties>
              <properties>
                <property>
                  <name>remoteAccessPointSpec</name>
                  <value>//localhost:12478/domainId=TDOM1_ID</value>
                </property>
              </properties>
            </connection-properties>
          </connection-instance>
          <connection-instance>
            <jndi-name>eis/TuxedoConnectionFactory2</jndi-name>
            <connection-properties>
              <properties>
                <property>
                  <name>spSecurity</name>
                  <value>APP_PW</value>
                </property>
                <property>
                  <name>applicationPassword</name>
                  <value>{Salted-AES}hHAsW13whgqTobG1t9Q92Q==</value>
                </property>
      <property>
        <name>remoteAccessPointSpec</name>
        <value>//localhost:12488/domainId=TDOM2_ID</value>
      </property>
    </properties>
  </connection-properties>
</connection-instance>
<connection-instance>
  <jndi-name>eis/TuxedoConnectionFactory3</jndi-name>
  <connection-properties>
    <properties>
      <property>
        <name>spSecurity</name>
        <value>DM_PW</value>
      </property>
      <property>
        <name>localAccessPointSpec</name>
        <value>//localhost:10801/domainId=JDOM_ID</value>
      </property>
      <property>
        <name>remoteAccessPointSpec</name>
        <value>//localhost:12498/domainId=TDOM3_ID,
        lPasswd1={Salted-AES}xNgOdUuXB7Z49D0cssluxA==,
        rPasswd1={Salted-AES}hAIzbPI+YyaeuHX0A9Umqg==</value>
      </property>
    </properties>
  </connection-properties>
</connection-instance>
```

```
    </connection-definition-group>
  </outbound-resource-adapter>
</weblogic-connector>
```

WebSphere Server

Normally, user configures the Oracle Tuxedo JCA Adapter after it has been installed.   User can configure RADD based adapter-wise property first and then configure factory property.

Configure Deployment Descriptor Property

The following is the procedure to configure adapter-wise property in the RADD from WebSphere Console.

1. Install Oracle Tuxedo JCA Adapter from console.

   Resources ' Resource Adapters ' Resource adapters

2. Click on the "name" column of the installed Oracle Tuxedo JCA Adapter.

   Assume you give the name "Tuxedo JCA Adapter"

3. From Resources ' Resource Adapters ' Resource adapter ' Tuxedo JCA Adapter.

   Click on "Custom properties" on the right hand side under "Additional Properties".

4. From Tuxedo JCA Adapter ' Custom properties

   Click on "Preference" on top of custom property table. This will cause the console expanded to add an input field "Maximum rows", change it from its default value to "60", and then click on "Apply" button right below the input field box and check box.

   Now user should be able to see all the configurable adapter-wise properties.

5. Clicks on the desired property name under the "Name" column.

   For instance: `"remoteAccessPointSpec"`

6. From Tuxedo JCA Adapter ' Custom properties ' `remoteAccessPointSpec`

   Change or add the desired `remoteAccessPointSpec` value to the "Value" field, then click on the "Apply" button and the click on "Save".

7. Use the same steps from step #5 to step #6 to modify another property.

If user configures `applicationPassword` property for WebSphere 7.0 user should not encrypt the password using `com.oracle.tuxedo.tools.EncryptPassword` tool because WebSphere 7.0 will encrypt the password.

Configure Factory Property

User uses the following steps to configure factory-based configuration for WebSphere.

1. Install Oracle Tuxedo JCA Adapter from console if it has not yet been done. (Resources ' Resource Adapters ' Resource adapters)
2. Click on the "name" column of Oracle Tuxedo JCA Adapter.

   Resources ' Resource Adapters ' Resource adapters

   Assumes you give name "Tuxedo JCA Adapter".
3. From Resources ' Resource Adapters ' Resource adapters ' Tuxedo JCA Adapter

   Click on "J2C connection factories" on the right hand side under "Additional Properties".
4. From Resources ' Resource Adapters ' Resource adapters ' Tuxedo JCA Adapter ' J2C connection factories

   Click on button "New".
5. From J2C connection factories ' New

   Enter the connection factory name in the "Name" field, and then enter the unique JNDI name in the "JNDI name" field, then click on "Apply" button at bottom of the page.

   Assume you entered "factory1" in the "Name" field.
6. From J2C connection factories ' factory1

   Click on the "Save" button. This will take you back to "J2C connection factories" page.
7. From Tuxedo JCA Adapter ' J2C connection factories

   Click on the "factory1" in the "Name" column, and then it will take you to page "factory1".
8. From Tuxedo JCA Adapter ' J2C connection factories ' factory1

   Click on the "Custom properties" in the right hand side under "Additional Properties".
9. From Tuxedo JCA Adapter ' J2C connection factories ' factory1 ' Custom properties

   Clicks on the property name in the "Name" column and then it will take you to property page. Assume you selected `"localAccessPointSpec"`.

10. From J2C connection factories ' factory1 ' Custom properties ' `localAccessPointSpec`

    Enter the desired value in the "Value" field, then click on "Apply" button, and then click on "Save". This will take you back to "Custom properties" page.

11. Repeat steps #9 and #10 for each property that need to be changed.

Oracle Tuxedo JCA Adapter Deployment

Deployment of the Resource Adapter is Application Server dependent. This is usually achieved by:

- Dropping the `.rar` file in a generic auto-deployment location
- Explicitly deploying the `.rar` file (or the directory containing its exploded version) via a console function
- Using an application supported scripting tool to install and deploy. For example `wsadmin` in WebSphere or `wldeployer` in WebLogic.
- Using system class path to load VIEW and FML classes, so it is also necessary to configure system class path.
- Using other web or application deployment mechanisms specified by the Application Server

Deploying the Oracle Tuxedo JCA Adapter involves choosing a deployment mode, configuring the resource adapter descriptor, repackaging the adapter and deploying it on a JCA 1.5 compliant JEE application server. In most cases, you only need to modify the `ra.xml` file (in the `META-INF` directory) to get the adapter up and running.

The section addresses the following topics:

- Oracle JCA Adapter Deployment Tasks
- Repackaging the Oracle Tuxedo JCA Adapter
- Changing the Connector Connection Pool Size.

Notes: In most cases, you only need to modify the `ra.xml` file (in the `META-INF` directory) to get the adapter up and running.

The resource archive configuration and repackaging procedure is the same regardless the type of the targeted application; however, the deployment of the Oracle Tuxedo JCA Adapter is different from application server to application server.

For more information, see your target application server documentation.

Oracle JCA Adapter Deployment Tasks

The following tasks are required in order to deploy the Oracle JCA Adapter:

1. Unjar the `com.oracle.tuxedo.TuxedoAdapter.rar` file into a directory.
2. Decides whether to use Client-Side only or full blown server and client.
   a. Choose to use Client-Side only
      - remove META-INF/ra.xml
      - remove META-INF/server.ra.xml
      - remove META-INF/sample.weblogic-ra.xml
      - rename META-INF/client-side.ra.xml to META-INF/ra.xml
      - modifies META-INF/ra.xml by adding properties for the /Domain configuration.

   b. Choose to use full blown server and client operations
      - remove META-INF/ra.xml
      - remove META-INF/client-side.ra.xml
      - remove META-INF/sample.weblogic-ra.xml
      - rename META-INF/server.ra.xml to META-INF/ra.xml
      - modify ra.xml file with desired none /Domain configuration properties, 'dmconfig' property must be specified
      - modify the dmconfig.xml with the desired and correct /Domain configuration
      - rename the dmconfig.xml file to whichever name 'dmconfig' property specified.
      - if you choose the 'dmconfig' file not to be treated as part of resource archive then move it to desired directory.

   c. Choose to use Factory-Based Configuration
      - remove META-INF/client-side.ra.xml
      - remove META-INF/server.ra.xml
      - remove META-INF/weblogic-ra.xml

- rename META-INF/sample.weblogic-ra.xml to META-INF/weblogic-ra.xml if configuring it for WebLogic server.
- For WebLogic server user modifies the META-INF/weblogic-ra.xml with the desired configuration. For WebSphere server user use the steps described in previous chapter to configure it from console.

3. jar the working directory to create Resource Adapter Archive
4. deploy the Resource Adapter Archive with the preferred method user want to use of the application server.
5. some application server may need to configure connection pool information through the console
6. some application server may need to 'activate' the adapter

Configuring Oracle Tuxedo JCA Adapter Deployment

The Oracle Tuxedo JCA Adapter checks if the class of the configured resource adapter is `TuxedoResourceAdapter`, or `TuxedoClientSideResourceAdapter`, or `TuxedoFBCResourceAdapter`.

TuxedoResourceAdapter Class

If `TuxedoResourceAdapter` is used, then the `dmconfig` property must be configured. If the configured `dmconfig` property contains only file name without any path information, the configuration is loaded as a resource as shown in Listing 32.

If it fails to load from the resource archive, it is treated as a file located in the current working directory. If the file fails to open, the resource adapter will not start and a `ResourceAdapterException` is thrown.

Listing 32 dmconfig Custom Property to Loaded dmconfig as a Resource

```
<config-property>
  <config-property-name>dmconfig</config-property-name>
   <config-property-type>java.lang.String</config-property-type>
   <config-property-value>dmconfig.xml</config-property-value>
</config-property>
```

If the dmconfig property configured contains path information, it is treated and loaded as file as shown in Listing 33. If the file fails to open, the resource adapter will not start and a ResourceAdapterException is thrown.

Listing 33 dmconfig Property Loaded as a File

```
…
<config-property>
   <config-property-name>dmconfig</config-property-name>
   <config-property-type>java.lang.String</config-property-type>
   <config-property-value>/user/dilbert/tja/dmconfig.
    xml</config-property-    value>
</config-property>
…
```

If the dmconfig file does not have any LocalAccessPoint configured then it creates a single default LocalAccessPoint. This default LocalAccessPoint can only have a session with RemoteAccessPoint using the default SessionProfile; it can only initiate outbound connection.

The Adapter creates a default SessionProfile using all the default values (except for the ConnectionPolicy which is always ON_STARTUP for the default SessionProfile. If SessionProfile is configured in the dmconfig file, it is constructed in addition to the default SessionProfile.

If the dmconfig file does not have SessionProfile configured, the adapter creates RemoteAccessPoint Sessions if the default LocalAccessPoint is created.

The resourceadapter-class element in the resource descriptor ra.xml file should contain the com.oracle.tuxedo.adapter.TuxedoResourceAdapter fully qualified class name as its value as show in Listing 34.

Listing 34 resourceadapter-class Element - com.oracle.tuxedo.adapter.TuxedoResourceAdapter

```
…
<resourceadapter>
```

```
  <resourceadapter-class>com.oracle.tuxedo.adapter.
   TuxedoResourceAdapter</resourceadapter-class>
…
```

Note: The new "client-side" mode properties are not available to this class-based resource adapter. If you configure these properties in the resource adapter deployment descriptor file, the behavior is application server dependent.

TuxedoClientSideResourceAdapter Class

If `TuxedoClientSideResourceAdapter` is configured, then `dmconfig` configuration is ignored. When this class of resource adapter is configured it assumes all the configuration information is in the resource adapter Java Bean provided by the application server JCA container.

If no `localAccessPointSpec` property is configured, a default `LocalAccessPoint` is created for the resource adapter deployment descriptor file-based configuration. The "at least one `LocalAccessPoint` must be configured" restriction in the 11gR1 11.1.1.1.0 release is removed.

If `remoteAccessPointSpec` property is configured, it is used to construct `RemoteAccessPoint`. If there is no `remoteAccessPointSpec` property configured, the configuration cannot be used and a warning message is logged in the adapter log file.

A default `SessionProfile` is created using information from properties related to Session Profile. If no session profile related properties are configured, the default `SessionProfile` is constructed using only the default values. It creates sessions from the `LocalAccessPoint` to every `RemoteAccessPoint` using the default `SessionProfile`.

The `resourceadapter-class` element in the resource descriptor `ra.xml` file, should contain the `com.oracle.tuxedo.adapter.TuxedoClientSideResourceAdapter` fully qualified class name as its value as shown in Listing 35.

Listing 35 resourceadapter-class Element - com.oracle.tuxedo.adapter.TuxedoClientSideResourceAdapter

```
…
<resourceadapter>
  <resourceadapter-class>com.oracle.tuxedo.adapter.
   TuxedoClientSideResourceAdapter</resourceadapter-class>
```

…

TuxedoFBCResourceAdapter Class

If `TuxedoFBCResourceAdapter` is configured, then dmconfig configuration and Resource Adapter Deployment Descriptor specific configuration properties are ignored. When this class of resource adapter is configured it assumes all the configuration information is in the resource adapter Java Bean provided by the application server JCA container and the Managed Connection Factory Java Bean that is also provided by the application server JCA container.

If no `localAccessPointSpec` property is configured for a factory, a default `LocaAccessPoint` is created to that factory. A `remoteAccessPointSpec` must be configured for each factory, and they will be used to construct `RemoteAccessPoint`.

A default `SessionProfile` is created for each factory using information from the properties related to `SessionProfile` of that factory. If no session profile related properties are configured for a factory, the factory will use default `SessionProfile`.

The `resourceadapter-class` element in the resource deployment descriptor `ra.xml` file, should contain the `com.oracle.tuxedo.adapter.TuxedoFBCResourceAdapter` fully qualified class name as its value as shown in Listing 36.

Listing 36 Resource Deployment Descriptor ra.xml File resourceadapter-class Element

```
...
<resourceadapter>
   <resourceadapter-class>com.oracle.tuxedo.adapter.
    TuxedoFBCResourceAdapter</resourceadapter-class>
...
```

Resource Adapter Deployment Descriptor Properties

There is a set of resource adapter deployment descriptor custom properties that are available to both resource adapter deployment descriptor based configuration and dmconfig based configuration. The only exception is "dmconfig" custom property that is only available using dmconfig based configuration; if user specified this property while using resource adapter deployment descriptor based configuration then this property will be ignored.

Customizing Properties

Some of the ra.xml file fields should not be changed as they pertain to the Oracle Tuxedo JCA Adapter internally or its descriptive information; however, there are other fields you must modify to customize the operation and application.

config-property

The config-property field is generally used to define Oracle Tuxedo JCA Adapter custom properties in the standard JCA deployment descriptor META-INF/ra.xml file. Table 28 lists the properties that are used to customize the Oracle Tuxedo JCA Adapter.

You must specify the Oracle Tuxedo JCA Adapter configuration file using the dmconfig property in the resource deployment descriptor META-INF/ra.xml file. For more information, see Configuration File Examples in the Oracle Tuxedo JCA Adapter Programming Guide.

Table 28 Customization Properties

Property	Type	Initial value (or default if not specified)	Description
traceLevel	java.lang.String	0	Level of debug tracing. For more information, see Transaction Support.
xaAffinity	java.lang.String	true	Turn on transaction-specific routing, for enhancing transaction performance.
keyFileName	java.lang.String	c:\tuxedo\keyfile	For dmconfig-based configuration only, this is the full path name to the key file. This file contains key used to encrypt all the passwords in the "dmconfig" file.
dmconfig	java.lang.String	C:\tuxedo\config.xml	For dmconfig-based configuration only, this is the full path name to the Oracle Tuxedo JCA Adapter configuration file. **Note:** If it does not contain path information then it will be treated as resource

Table 28 Customization Properties

Property	Type	Initial value (or default if not specified)	Description
appManagedLocalTxTimeout	java.lang.Integer	default value 300 seconds	Transaction timeout value of the local transaction managed by the Oracle Tuxedo JCA Adapter.
throwFailureReplyException	java.lang.Boolean	default value true	Customize CCI execution interface to decide whether to throw exception or failure reply data in the output data record in case an Oracle Tuxedo service returns failure reply.
autoTran	java.lang.Boolean	no default value	Enables/disables AUTOTRAN. By default there is no AUTOTRAN.

Trace Level Support

Table 29 lists the trace-level control values.

Table 29 Trace Level Control Values

Values	Components Traced	Description
20000	GWT_IO	Gateway input and output, including the ATMI verbs.
25000	GWT_EX	More Gateway information.
50000	JATMI_IO	JATMI input and output, including low-level JATMI calls.
55000	JATMI_EX	More JATMI information.
100000	All components	Information on all Oracle Tuxedo JCA Adapter components.

Note: Tracing is not executed for trace levels less that 20,000.

Listing 37 shows an example deployment descriptor file using the customization properties.

Listing 37 Customized Deployment Descriptor File Example

```
<config-property>
  <config-property-name>traceLevel</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>0</config-property-value>
</config-property>
<config-property>
  <config-property-name>xaAffinity</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>true</config-property-value>
</config-property>
<config-property>
  <config-property-name>keyFileName</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
<config-property-value>foo.key</config-property-value>
</config-property>
<config-property>
  <config-property-name>dmconfig</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
<config-property-value>dmconfig.xml</config-property-value>
</config-property>
<config-property>
  <config-property-name>throwFailureReplyException</config-property-name>
  <config-property-type>java.lang.Boolean</config-property-type>
<config-property-value>true</config-property-value>
</config-property-value>
</config-property>
<config-property>
  <config-property-name>appManagedLocalTxTimeout</config-property-name>
  <config-property-type>java.lang.Integer</config-property-type>
  <config-property-value>15</config-property-value>
</config-property>
```

Repackaging the Oracle Tuxedo JCA Adapter

Most application servers allow the resource adapter to be deployed in un-archived form; however, it is best to repackage the adapter after modifying the resource deployment descriptor before it is deployed.

Repackaging requires converting the Oracle Tuxedo JCA Adapter into a resource archive using the `"jar"` command that comes with the JDK. The resource archive has the `".rar"` extension.

For example, use the following command from the root directory of resource archive:

```
jar -cvf ../com.oracle.tuxedo.TuxedoAdapter.rar *
```

Changing the Connector Connection Pool Size

The application server has a default connection pool size. In most applications, the default connection pool size is large enough; however, in some situations the default connection pool size may not be enough. For example, the Oracle WebLogic Server has a default connection pool size equal to 10 which means it can support a maximum of 10 concurrent JCA clients using the same adapter to access remote Enterprise Information Systems (EIS).   If an application wants to support more than 10 concurrent clients using the Oracle Tuxedo JCA adapter, then the application must expand the connection pool size.

For example, to change the connection pool size from default value to 20 in an Oracle WebLogic Server installation, you can modify the `weblogic-ra.xml` file in the `META-INF` directory as shown in Listing 38.

Listing 38   Change Oracle WebLogic Connector Connection Pool Size

```
 <outbound-resource-adapter>
  <default-connection-properties>
    <pool-params>
      <initial-capacity>15</initial-capacity>
      <max-capacity>20</max-capacity>
    </pool-params>
  </default-connection-properties>
  <connction-definition-group>
  ...
  </connection-definition-group>
</outbound-resource-adapter>
```

Note: Different application servers have require different connection pool size configuration. Some application servers use the custom deployment descriptor to configure connection pool size (such as Oracle WebLogic Server). Other application servers use console configuration (such as IBM WebSphere).

For more information, see your application server documentation.

Oracle Tuxedo JCA Adapter Management

After deploying the Oracle Tuxedo JCA Adapter, you may want to expand or change the application. To do so, you must modify the adapter configuration. Modifying the Oracle Tuxedo JCA Adapter configuration involves using a text editor or XML editor to alter the contents of the configured adapter configuration.

For Oracle WebLogic Server and JBOSS application server environments, you must do the following steps:

1. Modify the configuration file
2. Stop the adapter
3. Restart the adapter with the new configuration file

Note: For IBM WebSphere application servers, you must stop the application server and then restart it.

Runtime Configuration Updating

The Oracle Tuxedo JCA Adapter does not support runtime dynamic configuration modification without stopping and restarting for the configuration changes to take effect.

Oracle Tuxedo JCA Adapter Fail Over Capability

Oracle Tuxedo JCA Adapter supports both link level fail over and service level fail over.

The link level fail over gives user ability to specify different Access Point network addresses. This is applicable to both Local Access Point and Remote Access Point. In dmconfig file configuration user can specify multiple `NetworkAddress` element in the RemoteAccessPoint and LocalAccessPoint. The order of these network address dictate the order of preference. The Oracle Tuxedo JCA Adapter will attempt using the first network address in the LocalAccess Point to establish listening endpoint, if it failed it will try the next one until either listening endpoint established or it exhausted all the network addresses. The Oracle Tuxedo JCA Adapter will attempt to use the first network address to establish connection with remote Tuxedo GWTDOMAIN gateway, if it failed it will try the next one until either connection established or it exhausted all the network addresses. However, there is no link level fail back. The following example shows using dmconfig file configuration that a local access point, LDOM1, has 2 network addresses for link level failover; and it also shows that a remote access point, RDOM1, has 2 network addresses for link level fail over.

Listing 39 Link Level Fail Over Example

```
...
<LocalAccessPoint name="LDOM1">
   <AccessPointId>Godfried</AccessPointId>
   <NetworkAddress>//neocortex:14001</NetworkAddress>
   <NetworkAddress>//cerebrum:14002</NetworkAddress>
</LocalAccessPoint>
<RemoteAccessPoint name="RDOM1">
   <AccessPointId>Geneve</AccessPointId>
   <NetworkAddress>//bluestar:11023</NetworkAddress>
   <NetworkAddress>//orion:11023</NetworkAddress>
</RemoteAccessPoint>
...
```

Besides link level fail over Oracle Tuxedo JCA Adapter also supports Service Level Fail Over. The service level fail over specifies the alternate sessions that the remote Tuxedo resource can be accessed. This is a comma separated list. It is different from load balancing. Service Level Fail Over only available when ConnectionPolicy equals to ON_STARTUP. Other types of ConnectionPolicy will only have load balancing. When Service level fail over is enabled it will check the session status of the primary session, if its status is not connected it will check the 1st backup session and so on until either a connected status found for a backup session or it exhausted all the configured backup session.

The following example shows an imported resource, TOUPPER, can be load balancing between session_1 and session_2.

Listing 40 Load Balancing Example

```
...
<Import name="TUXTOUPPER">
  <SessionName>session_1</SessionName>
  <SessionName>session_2</sessionName>
  <LoadBalancing>RoundRobin</LoadBalancing>
</Import>
...
```

The following example shows not only it is capable of load balancing but also capable of service level fail over.

Listing 41 Load Balancing with Service Level Fail Over Example

```
...
 <Import name="TUXTOUPPER">
   <SessionName>session_1,session_3</SessionName>
   <SessionName>session_2,session_3</sessionName>
   <LoadBalancing>RoundRobin</LoadBalancing>
</Import>
 ...
```

The above configuration will load balancing between session_1 and session_2. In the event that session_1 is not available the service request will be forwarded to session_3 when load balancing algorithm decided it is session_1's turn; and the same for session_2, in this case session_3 also backs up session_2.

The service level fail back is automatic when `ConnectionPolicy` is set to `ON_STARTUP` for all the sessions of a particular imported resource. With the above example if session_1 is not available then all the service request destined to session_1 will be routed to session_3; when session_1 become available then service request will be routed back to the session_1 the primary route.

Oracle Tuxedo JCA Adapter Security

The Oracle Tuxedo JCA Adapter supports data security using either Link-Level Encryption or Secured Socket Layer. It also provides outbound identity propagation from application servers to Oracle Tuxedo. This provides finer grain control over Oracle Tuxedo resource access.

The "`DMConfigChecker`" utility is used to ensure security. This utility not only checks the configuration file against the schema, but also converts the password into an encrypted form for better security. If encryption is required, you must run `DMConfigChecker` before starting the Oracle Tuxedo JCA Adapter. For more information, see the Oracle Tuxedo JCA Reference Guide.

Link-Level Encryption (LLE)

Link-Level Encryption (LLE), is a fast, proprietary technology that encrypts all user message flow between the Oracle Tuxedo JCA Adapter and the Tuxedo TDomain gateway. It supports 40-bit, 56-bit and 128-bit encryption strength. This feature is enabled by configuring `MaxEncryptBits` and `MinEncryptBits` in the `SessionProfile` of the adapter configuration.

The default value for `MaxEncryptBits` is 128-bit and the default value for `MinEncryptBits` is 0. The permissible values for both elements are 0 (no encryption), 40-bit, 56-bit, 128-bit, and 256-bit. 256-bit encryption is for SSL AES 256-bit encryption (LLE does not support 256-bit encryption). The `MinEncryptBits` value must be smaller than or equal to the `MaxEncryptBits` value.

Notes: If SSL is not configured and `MaxEncryptBits` is set to256-bit, `MaxEncryptBits` is scaled down to maximum 128-bit LLE.

LLE must also be configured in the `GWTDOMAIN` gateway.

Secured Socket Layer (SSL) Encryption

The Oracle Tuxedo JCA Adapter also support Secure Socket Layer (SSL) encryption. To enable SSL encryption, you must do the following:

- Configure Session Profile `MinEncryptBits` and `MaxEncryptBits`.
- Configure Local Access Point `SSLInfo`.

Java Key Store (JKS) is supported. Both "`IdentityKeyStoreFileName`" and "`TrustKeyStoreFileName`" points to the JKS type.

Note: By default, mutual authentication is disabled; however, it can be enabled it setting the "MutualAuthenticationRequired" element to "true".

An sample SSLInfo configuration example is shown in Listing 42.

Listing 42 SSLInfo Configuration Example

```
<LocalAccessPoint name="jdom">
...
  <SSLInfo>
<IdentityKeyStoreFileName>c:\test\cert\test_users.jks</IdentityKeyStoreFil
eName>
<IdentityKeyStorePassPhraseEncrypted>passphrase</IdentityKeyStorePassPhras
eEncrypted>
    <PrivateKeyAlias>tester</PrivateKeyAlias>
<PrivateKeyPassPhraseEncrypted>passphrase</privatekeypassphraseencrypted>
    <TrustKeyStoreFileName>c:\test\cert\trusted.jks</TrustKeyStoreFileName
>
<TrustKeyStorePassPhraseEncrypted>passphrase</TrustKeyStorePassPhraseEncry
pted>
  </SSLInfo>
</LocalAccessPoint>
```

In this example, you must run the DMConfigChecker utility before it can be used by the Oracle Tuxedo JCA Adapter (since three elements require encryption).

Session Authentication

The Oracle Tuxedo JCA Adapter supports session authentication when SSL is not configured, but the "Security" of the SessionProfile must be configured using "DM_PW" or "APP_PW".

When the security is configured using "APP_PW", it uses the Oracle Tuxedo Application Password as a key to encrypt/decrypt the authenticator. Only one Application Password can be configured for the Oracle Tuxedo JCA Adapter. It is configured in the Resources "ApplicationPasswordEncrypted" element in the configuration file.

When security is configured with "DM_PW", it uses the Domain Password as a key to encrypt/decrypt the authenticator. At most, two passwords pairs can be configured for any given session configured.

Each password pair consists of one local password and one remote password and their activation/deactivation time. The activation/deactivation time uses the format "YYYY:MM:DD:hh:mm:ss" (for example: 2009:01:01:12:00:00).

The deactivation time must be later than the activation time. If the activation time is not specified it indicates that the password is already activated when the adapter is booted. If the deactivation time is not specified, then the password never expires.

If a password pair expired while the session already established, it will not invalidate the session and starts the session negotiation process. However, if a password pair expired *before* the session negotiation started, then that password pair is not be used for authentication. If no valid password pair is found during session authentication, then the session cannot be established.

If SSL is required for a Local Access Point, then the session between that Local Access Point and any Remote Access Point uses SSL to underline the data privacy mechanism. In this case, Session Authentication is not be performed even if "SessionProfile/Security" is configured with the correct values.

Appkey Generator

The AppKey Generator is a pluggable class that is used to determine user information to be sent from the application server to Tuxedo. Oracle Tuxedo uses this information to determine user access rights to a Tuxedo resource. This is also called ACL.

The Oracle Tuxedo JCA Adapter comes preconfigured with a default AppKey Generator class to work with default Oracle Tuxedo Authentication, Authorization, and Auditing plug-in (AAA). However, if Access Control is required using the default Oracle Tuxedo AAA plug-in, then you must also configure the TpusrFile element in the Remote Access Point. The TpusrFile element in RemoteAccessPoint points to an Oracle Tuxedo "tpusr" file.

The easiest way to do this is to copy it from Oracle Tuxedo or share it with Oracle Tuxedo if both are running on the same machine. If the RemoteAccessPoint TpusrFile file is not configured, the Oracle JCA Adapter also looks for the TpusrFile file in the Resources section.

You must also set "CredentialPolicy" to "Global" in order to allow the AAA security token to move across the network from the Oracle Tuxedo JCA Adapter to an Oracle Tuxedo application domain.

There are other configuration elements in the `RemoteAccessPoint` that further give you the ability to customize the AppKey Generator plug-in. The "`AllowAnonymous`" element tells adapter whether or not anonymous access to Oracle Tuxedo is allowed.

Note: By default, anonymous access is not allowed; the application server performs authentication.

The "`DefaultApplicationKey`" is the key value used by anonymous users to access Oracle Tuxedo (the default value is "`-1`"). The default AppKey Generator assumes the anonymous user name is "anonymous".

In order to have successful identity propagation (in addition to all the previously described configuration options), you must also configure the "`Principal Mapping`" element in the host application server JCA container.

For more information, see your target application server documentation.

An example Oracle JCA Adapter configuration file identity propagation example is shown in Listing 43.

Listing 43 Identity Propagation Example

```
<RemoteAccessPoint name="tdom">
  ...
  <TpusrFile>c:\test\data\tpusr</TpusrFile>
  <AllowAnonymous>true</AllowAnonymous>
  ...
</RemoteAccessPoint>
<SessionProfile name="prof_1">
  ...
  <CredentialPolicy>global</CredentialPolicy>
  ...
</SessionProfile>
```

See Also

- Oracle Tuxedo JCA Adapter Programming Guide
- Oracle Tuxedo JCA Adapter Reference Guide

- Oracle Tuxedo Reference Guide DMCONFIG and GWTDOMAIN