Name create-connector-connection-pool– adds a connection pool with the specified connection pool name

Synopsis

```
create-connector-connection-pool [--help] [--target=target]
--raname raname
--connectiondefinition connectiondefinitionname
[--steadypoolsize steadypoolsize]
[--maxpoolsize maxpoolsize]
[--maxwait maxwait]
[--poolresize poolresize]
[--idletimeout idletimeout]
[--isconnectvalidatereq={false|true}]
[--failconnection={false|true}]
[--leaktimeout=timeout]
[--leakreclaim={false|true}]
[--creationretryattempts=attempts]
[--creationretryinterval=interval]
[--lazyconnectionenlistment={false|true}]
[--lazyconnectionassociation={false|true}]
[--associatewiththread={false|true}]
[--matchconnections={true|false}]
[--maxconnectionusagecount=count]
[--validateatmostonceperiod=interval]
[--transactionsupport transactionsupport]
[--description text]
[--property (name=value)[:name=value]*]
poolname
```

Description The `create-connector-connection-pool` subcommand defines a pool of connections to an enterprise information system (EIS). The named pool can be referred to by multiple connector resources. Each defined pool is instantiated at server startup, and is populated when accessed for the first time. If two or more connector resources point to the same connector connection pool, they are using the same pool of connections at run time. There can be more than one pool for a connection definition in a single resource adapter.

A connector connection pool with authentication can be created either by using a `--property` option to specify user, password, or other connection information, or by specifying the connection information in the XML descriptor file.

This subcommand is supported in remote mode only.

Options

`--help`
`-?`

Displays the help text for the subcommand.

`--target`

Do not specify this option. This option is retained for compatibility with other releases. If you specify this option, a syntax error does not occur. Instead, the subcommand runs successfully and the option is silently ignored.

`--raname`
The name of the resource adapter.

`--connectiondefinition`
The name of the connection definition.

`--steadypoolsize`
The minimum and initial number of connections maintained in the pool. Default value is 8.

`--maxpoolsize`
The maximum number of connections that can be created to satisfy client requests. Default value is 32.

`--maxwait`
The amount of time, in milliseconds, that a caller must wait before a connection is created, if a connection is not available. If set to 0, the caller is blocked indefinitely until a resource is available or until an error occurs. Default value is 60000.

`--poolresize`
Quantity by which the pool will scale up or scale down the number of connections. Scale up: When the pool has no free connections, pool will scale up by this quantity. Scale down: All the invalid and idle connections are removed, sometimes resulting in removing connections of quantity greater than this value. The number of connections that is specified by `--steadypoolsize` will be ensured. Possible values are from 0 to `MAX_INTEGER`. Default value is 2.

`--idletimeout`
The maximum time that a connection can remain idle in the pool. After this amount of time, the pool can close this connection. Default value is 300.

`--isconnectvalidatereq`
If the value is set to true, the connections will be checked to see if they are usable, before they are given out to the application. Default value is false.

`--failconnection`
If set to true, all connections in the pool are closed if a single validation check fails. This parameter is mandatory if the `--isconnectvalidatereq` option is set to true. Default value is false.

`--leaktimeout`
Specifies the amount of time, in seconds, for which connection leaks in a connection pool are to be traced.

If connection leak tracing is enabled, you can use the Admin Console to enable monitoring of the JDBC connection pool to get statistics on the number of connection leaks. Default value is 0, which disables connection leak tracing.

`--leakreclaim`
Specifies whether leaked connections are restored to the connection pool after leak connection tracing is complete. Possible values are as follows:

`false`
Leaked connections are *not* restored to the connection pool (default).

`true`
Leaked connections are restored to the connection pool.

`--creationretryattempts`
Specifies the maximum number of times that the server retries to create a connection if the initial attempt fails.

Default value is 0, which specifies that the server does not retry to create the connection.

`--creationretryinterval`
Specifies the interval, in seconds, between successive attempts to create a connection.

If `--creationretryattempts` is 0, the `--creationretryinterval` option is ignored. Default value is 10.

`--lazyconnectionenlistment`
Specifies whether a resource to a transaction is enlisted only when a method actually uses the resource. Possible values are as follows:

`false`
Resources to a transaction are always enlisted and *not* only when a method actually uses the resource (default).

`true`
Resources to a transaction are enlisted *only* when a method actually uses the resource.

`--lazyconnectionassociation`
Specifies whether a physical connection should be associated with the logical connection only when the physical connection is used, and disassociated when the transaction is completed. Such association and dissociation enable the reuse of physical connections. Possible values are as follows:

`false`
A physical connection is associated with the logical connection even before the physical connection is used, and is *not* disassociated when the transaction is completed (default).

`true`
A physical connection is associated with the logical connection only when the physical connection is used, and disassociated when the transaction is completed. The `--lazyconnectionenlistment` option must also be set to `true`.

`--associatewiththread`
Specifies whether a connection is associated with the thread to enable the thread to reuse the connection. If a connection is not associated with the thread, the thread must obtain a connection from the pool each time that the thread requires a connection. Possible values are as follows:

`false`
A connection is *not* associated with the thread (default).

`true`
A connection is associated with the thread.

`--matchconnections`
Specifies whether a connection that is selected from the pool should be matched with the resource adaptor. If all connections in the pool are identical, matching between connections and resource adapters is not required. Possible values are as follows:

`true`
A connection should be matched with the resource adaptor (default).

`false`
A connection should *not* be matched with the resource adaptor.

`--maxconnectionusagecount`
Specifies the maximum number of times that a connection can be reused.

When this limit is reached, the connection is closed. By limiting the maximum number of times that a connection can be reused, you can avoid statement leaks. Default value is 0, which specifies no limit on the number of times that a connection can be reused.

`--validateatmostonceperiod`
Specifies the time interval in seconds between successive request to validate a connection at most once. Setting this attribute to an appropriate value minimizes the number of validation requests by a connection. Default value is 0, which specifies that the connection is never validated.

`--transactionsupport`
Indicates the level of transaction support that this pool will have. Possible values are `XATransaction`, `LocalTransaction` and `NoTransaction`. This attribute can have a value lower than or equal to but not higher than the resource adapter's transaction support attribute. The resource adapter's transaction support attribute has an order of values, where `XATransaction` is the highest, and `NoTransaction` the lowest.

`--description`
Text providing descriptive details about the connector connection pool.

`--property`
Optional attribute name/value pairs for configuring the pool. The following properties are available:

`AddressList`
Specifies a list of host/port combinations of the Message Queue software. For JMS resources of the type `javax.jms.TopicConnectionFactory` or `javax.jms.QueueConnectionFactory`. There is no default value.

`ClientId`
Specifies the JMS Client Identifier to be associated with a Connection created using the `createTopicConnection` method of the `TopicConnectionFactory` class. For JMS resources of the type `javax.jms.TopicConnectionFactory`. There is no default value.

Durable subscription names are unique and only valid within the scope of a client identifier. To create or reactivate a durable subscriber, the connection must have a valid client identifier. The JMS specification ensures that client identifiers are unique and that a given client identifier is allowed to be used by only one active connection at a time. There is no default value.

`UserName`
Specifies the user name for connecting to the Message Queue software. For JMS resources of the type `javax.jms.TopicConnectionFactory` or `javax.jms.QueueConnectionFactory`. Default value is guest.

`Password`
Specifies the password for connecting to the Message Queue software. For JMS resources of the type `javax.jms.TopicConnectionFactory` or `javax.jms.QueueConnectionFactory`. Default value is guest.

`ReconnectAttempts`
Specifies the number of attempts to connect (or reconnect) for each address in the `imqAddressList` before the client runtime moves on to try the next address in the list. A value of −1 indicates that the number of reconnect attempts is unlimited (the client runtime attempts to connect to the first address until it succeeds). Default value is 6.

`ReconnectInterval`
Specifies the interval between reconnect attempts in milliseconds. This applies to attempts on each address in the `imqAddressList` and on successive addresses in the list. If too short, this time interval does not give a broker time to recover. If too long, the reconnect might represent an unacceptable delay. Default value is 30000 (30 seconds).

`ReconnectEnabled`
If true, specifies that the client runtime attempts to reconnect to a message server (or the list of addresses in `imqAddressList`) when a connection is lost. Default value is false.

`AddressListBehavior`
Specifies whether connection attempts are in the order of addresses in the `imqAddressList` attribute (`priority`) or in a random order (`random`). If many clients are attempting a connection using the same connection factory, use a random order to prevent them from all being connected to the same address. Default value is priority.

`AddressListIterations`
Specifies the number of times the client runtime iterates through the address list in an effort to establish (or reestablish) a connection. Default value is 1.

`LazyConnectionEnlistment`
Deprecated. Use the equivalent attribute. Default value is false.

`LazyConnectionAssociation`
Deprecated. Use the equivalent attribute. Default value is false.

`AssociateWithThread`
Deprecated. Use the equivalent attribute. Default value is false.

`MatchConnections`
Deprecated. Use the equivalent attribute. Default value is false.

Operands

poolname
The name of the connection pool to be created.

Examples

EXAMPLE 1 Creating a Connector Connection Pool

This example creates a new connector connection pool named `jms/qConnPool`.

```
asadmin> create-connector-connection-pool
--raname jmsra --connectiondefinition javax.jms.QueueConnectionFactory jms/qConnPool
--steadypoolsize 20 --maxpoolsize 100 --poolresize 2 --maxwait 60000
Command create-connector-connection-pool executed successfully
```

Exit Status

0	subcommand executed successfully
1	error in executing the subcommand

See Also

`delete-connector-connection-pool`(1), `list-connector-connection-pools`(1), `ping-connection-pool`(1)

`asadmin`(1M)