Name

create-jdbc-connection-pool– registers a JDBC connection pool

Synopsis

```
create-jdbc-connection-pool
[--terse={true|false}][ --echo={true|false} ]
[ --interactive={true|false} ] [ --host  host]
[--port port] [--secure| -s ] [ --user  admin_user]
[--passwordfile filename] [--help]
[ --target =target]
--datasourceclassname=classname [--restype=res_type]
[--steadypoolsize=poolsize] [--maxpoolsize=poolsize]
[--maxwait=time] [--poolresize=limit]
[--idletimeout=time] [--isolationlevel=isolation_level]
[--isisolationguaranteed={true|false}] [ --isconnectvalidatereq ={false|true}]
[ --validationmethod =method] [--validationtable=tablename]
[--failconnection={false|true}] [ --allownoncomponentcallers ={false|true}]
[ --nontransactionalconnections ={false|true}]
[ --leaktimeout =timeout] [--leakreclaim={false|true}]
[ --creationretryattempts =attempts]
[--creationretryinterval=interval] [--statementtimeout=timeout]
[--lazyconnectionenlistment={false|true}]
[ --lazyconnectionassociation ={false|true}]
[ --associatewiththread ={false|true}] [ --matchconnections ={false|true}]
[ --maxconnectionusagecount =count]
[--validateatmostonceperiod=interval]
[--wrapjdbcobjects={false|true}]
[ --description  text] [--property (name=value)
[:name=value]*] connectionpoolid
```

Description

The `create-jdbc-connection-pool` command registers a new Java™ DataBase Connectivity (“JDBC™”) software connection pool with the specified JDBC connection pool name.

This command is supported in remote mode only.

Options

`-t --terse`

Indicates that any output data must be very concise, typically avoiding human-friendly sentences and favoring well-formatted data for consumption by a script. Default is false.

`-e --echo`

If set to true, the command-line statement is echoed on the standard output. Default is false.

`-I --interactive`

If set to true (default), only the required password options are prompted.

`-H --host`

The machine name where the domain administration server is running. The default value is `localhost`.

`-p --port`
The HTTP port or HTTPS port for administration. This port is the port in the URL that you specify in your web browser to manage the domain, for example, `http://localhost:4848`.

The default port number for administration is 4848.

`-s --secure`
If set to true, uses SSL/TLS to communicate with the domain administration server.

The default is false.

`-u --user`
The user name of the authorized administrative user of the domain administration server.

If you have authenticated to a domain by using the `asadmin login` command, you need not specify the `--user` option for subsequent operations on the domain.

`--passwordfile`
Specifies the name, including the full path, of a file that contains the password entries in a specific format.

The entry for a password must have the `AS_ADMIN_` prefix followed by the password name in uppercase letters. For example, to specify the password for the domain administration server, use an entry with the following format:

`AS_ADMIN_PASSWORD=`*password*

In this example, *password* is the actual administrator password.

The following other passwords can also be specified:

- `AS_ADMIN_MAPPEDPASSWORD`
- `AS_ADMIN_USERPASSWORD`
- `AS_ADMIN_ALIASPASSWORD`

All remote commands must specify the administration password to authenticate to the domain administration server. The password can be specified by one of the following means:

- Through the `--passwordfile` option
- Through the `asadmin login` command
- Interactively at the command prompt

The `asadmin login` command can be used only to specify the administration password. For other passwords that remote commands require, use the `--passwordfile` option or specify them at the command prompt.

After authenticating to a domain by using the `asadmin login` command, you need not specify the administration password through the `--passwordfile` option for subsequent operations on the domain. However, only the `AS_ADMIN_PASSWORD` option is not required. You still must provide the other passwords, for example, `AS_ADMIN_USERPASSWORD`, when required by individual commands, such as `update-file-user`.

For security reasons, a password that is specified as an environment variable is not read by the `asadmin` command.

The default value for `AS_ADMIN_MASTERPASSWORD` is `changeit`.

`--help`
Displays the help text for the command.

`--target`
This option is deprecated.

`--datasourceclassname`
The name of the vendor-supplied JDBC datasource resource manager.

`--restype`
The interface that the datasource class implements. Must be one of `javax.sql.DataSource`, `javax.sql.ConnectionPoolDataSource` or `javax.sql.XADataSource`. It leads to an error when this option has a legal value and the indicated interface is not implemented by the datasource class. This option has no default value.

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

`--maxpoolsize`
The maximum number of connections that can be created. The default value is 32.

`--maxwait`
The amount of time a caller will wait before a connection timeout is sent. The default is 60 seconds. A value of 0 forces the caller to wait indefinitely.

`--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. Steadypoolsize will be ensured. Possible values are from 0 to `MAX_INTEGER`. The default value is 2.

`--idletimeout`
The maximum time, in seconds, that a connection can remain idle in the pool. After this time, the implementation can close this connection. This timeout value must be kept shorter than the server side timeout value to prevent the accumulation of unusable connections in the application. The default value is 300.

`--isolationlevel`
The transaction-isolation-level on the pooled database connections. This option does not have a default value. If not specified, the pool operates with the default isolation level that the JDBC driver provides.

You can set a desired isolation level using one of the standard transaction isolation levels: `read-uncommitted`, `read-committed`, `repeatable-read`, `serializable`. Applications that change the isolation level on a pooled connection programmatically risk polluting the pool. This could lead to program errors.

`--isisolationguaranteed`
This is applicable only when a particular isolation level is specified for transaction-isolation-level. The default value is true.

This option assures that every time a connection is obtained from the pool, isolation level is set to the desired value. This could have some performance impact on some JDBC drivers. Administrators can set this to false when the application does not change `--isolationlevel` before returning the connection.

`--isconnectvalidatereq`
If set to true, connections are validated or checked to see if they are usable before giving out to the application. The default value is false.

`--validationmethod`
The name of the validation table used to perform a query to validate a connection. Valid settings are: `auto-commit`, `meta-data`, or `table`. The default value is `auto-commit`.

`--validationtable`
The name of the validation table used to perform a query to validate a connection.

`--failconnection`
If set to true, all connections in the pool must be closed when a single validation check fails. The default value is false. One attempt is made to re-establish failed connections.

`--allownoncomponentcallers`
A pool with this property set to true can be used by non-Java EE components, that is, components other than EJBs or Servlets. The returned connection is enlisted automatically with the transaction context obtained from the transaction manager.

`--nontransactionalconnections`
A pool with this property set to true returns non-transactional connections. This connection does not get automatically enlisted with the transaction manager.

`--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 Administration Console to enable monitoring of the JDBC connection pool to get statistics on the number of connection leaks.

The 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 Enterprise Server retries to create a connection if the initial attempt fails.

The default value is 0, which specifies that Enterprise 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.

The default value is 10.

`--statementtimeout`
Specifies the length of time in seconds after which a query that is not completed is terminated.

A query that remains incomplete for a long period of time might cause the application that submitted the query to hang. To prevent this occurrence, use this option set a timeout for all statements that will be created from the connection pool that you are creating. When creating a statement, Enterprise Server sets the `QueryTimeout` property on the statement to the length of time that is specified.

The default value is -1, which specifies that incomplete queries are never terminated.

`--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:

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

`true`
A connection should 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.

The 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.

The default value is 0, which specifies that the connection is never validated.

`--wrapjdbcobjects`

Specifies whether the pooling infrastructure provides wrapped JDBC objects to applications.

By providing wrapped JDBC objects, the pooling infrastructure prevents connection leaks by ensuring that applications use logical connections from the connection pool, not physical connections. The use of logical connections ensures that the connections are returned to the connection pool when they are closed. However, the provision of wrapped JDBC objects can impair the performance of applications.

The pooling infrastructure provides wrapped objects for implementations of the following interfaces in the JDBC API:

- `java.sql.CallableStatement`
- `java.sql.DatabaseMetaData`
- `java.sql.PreparedStatement`
- `java.sql.ResultSet`
- `java.sql.Statement`

Possible values of `--wrapjdbcobjects` are as follows:

`false`

The pooling infrastructure does *not* provide wrapped JDBC objects to applications. (default).

`true`

The pooling infrastructure provides wrapped JDBC objects to applications.

`--description`

Text providing details about the specified JDBC connection pool.

`--property`

Optional attribute name/value pairs for configuring the connection pool.

Note – If an attribute name or attribute value contains a colon, the backslash (\) must be used to escape the colon in the name or value. Other characters might also require an escape character. For more information about escape characters in command options, see the `asadmin(1M)` man page.

Operands

connectionpoolid The name of the JDBC connection pool to be created.

Examples

EXAMPLE 1 Using create-jdbc-connection-pool command

```
asadmin> create-jdbc-connection-pool --user admin
--passwordfile passwords.txt --host localhost --port 7070
--datasourceclassname org.apache.derby.jdbc.ClientDataSource
--restype javax.sql.XADataSource
--property portNumber=1527:password=APP:user=APP:serverName=
localhost:databaseName=sun-appserv-samples:connectionAttributes=\;
create\\=true sample_derby_pool
Command create-jdbc-connection-pool executed successfully
```

Where, the `sample_derby_pool` is created. The escape character backslash (\) is used in the `--property` option to distinguish the semicolon (`;`). Two backslashes (\\) are used to distinguish the equal (=) sign.

Exit Status

0	command executed successfully
1	error in executing the command

See Also

`delete-jdbc-connection-pool`(1), `list-jdbc-connection-pools`(1)

`asadmin`(1M)