

Using datasource pool wizard in GEP

[Making deployment plan changes with Geronimo Deployment Plan Editor](#)

[Getting familiar with Geronimo Eclipse Plugin \(GEP\)](#)

[Using security realm wizard in GEP](#)

This article shows you how to create a datasource pool in GEP during development. The datasource pool wizard is a handy feature to speed up your development by creating a datasource connector without any manual coding. Information of the connector you create using this wizard is embedded in the deployment plan of the application as an `<ext-module>` element.

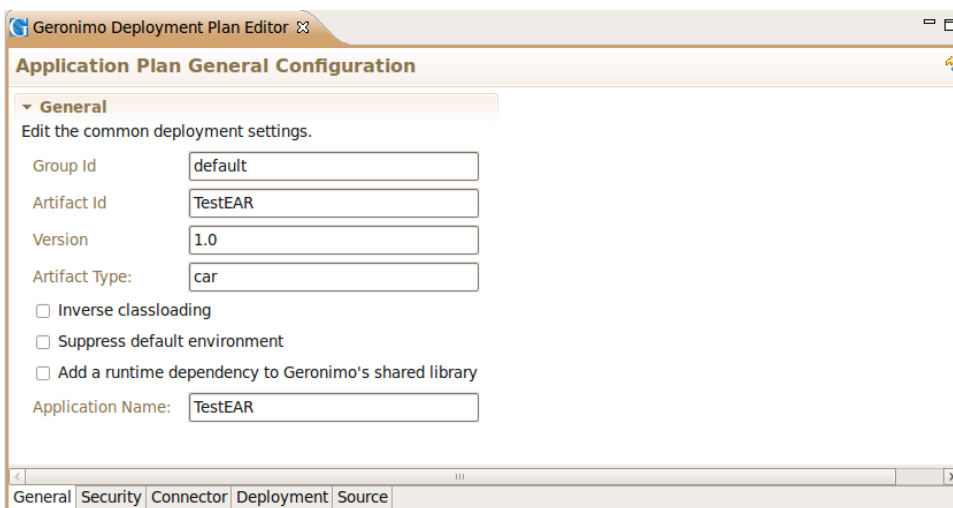
Note that this wizard has the following 2 limitations:

- It is only available when you develop a EAR project In Eclipse.
- It only supports the embedded Apache Derby database.

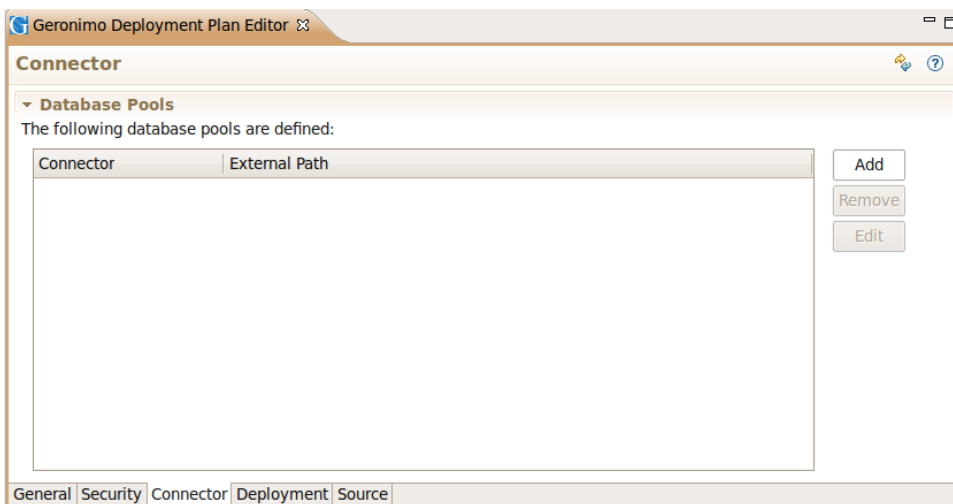
Adding a new datasource pool

To use the datasource pool wizard in GEP:

1. Double-click the `geronimo-application.xml` file under the `/META-INF` directory of your application to open it in the **Geronimo Deployment Plan Editor**. The following figure shows the **General** tab of the editor:



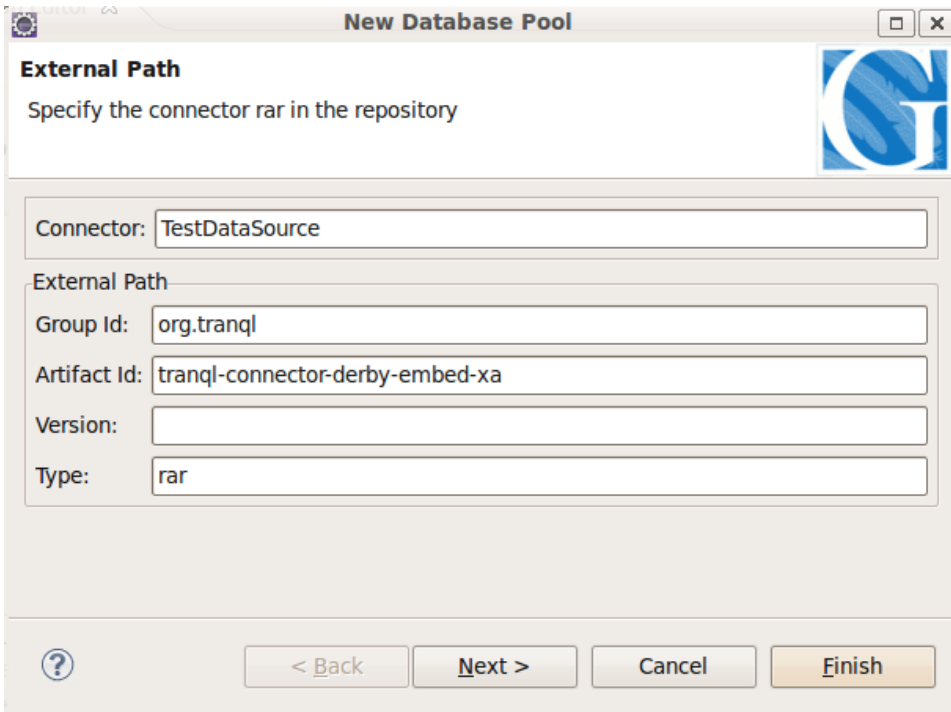
2. Click the **Connector** tab, and then open the **Database Pools** section to see the datasource pools that are defined. The following figure shows the **Database Pools** section in the **Connector** tab:



The following list describes the functions of the three buttons in the **Database Pools** section:

- **Add** - create a new datasource pool.
- **Remove** - remove an existing datasource connector.
- **Edit** - reconfigure an existing datasource connector.

3. Click **Add** to start a datasource pool wizard, and then fill in the required fields. The following figure shows the database pool wizard:



The screenshot shows the 'New Database Pool' wizard window. The title bar says 'New Database Pool'. The main heading is 'External Path' with the instruction 'Specify the connector rar in the repository'. There is a Geronimo logo in the top right. Below the heading, there are four input fields: 'Connector' with the value 'TestDataSource', 'Group Id' with 'org.tranql', 'Artifact Id' with 'tranql-connector-derby-embed-xa', and 'Type' with 'rar'. At the bottom, there are navigation buttons: a help icon, '< Back', 'Next >', 'Cancel', and 'Finish'.

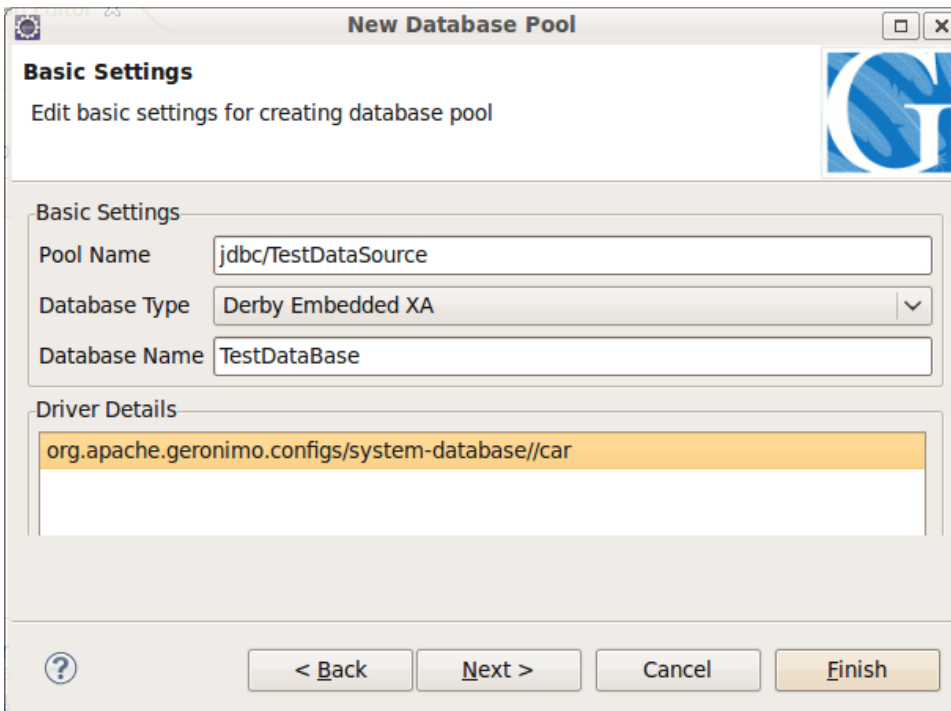
The following list introduces the fields in this step of the datasource pool wizard:

- **Connector:** specify the location of a Java EE Connector module.
- **External-path:** indicate that the module is not part of the enterprise application and must be located by matching the supplied pattern in a Geronimo repository.

4. Click **Next** to edit **Basic Settings** and choose a driver for the datasource pool.

The basic settings include pool name, database type and database name. Because this wizard only supports embedded Apache Derby database, there are only 2 available options(xa and non-xa) in the **Database Type** drop-down list and only one driver in the **Driver Details** field. The driver is `org.apache.geronimo.configs/system-database//car`.

The following figure shows the fields in this step:



The screenshot shows the 'New Database Pool' wizard window at the 'Basic Settings' step. The title bar says 'New Database Pool'. The main heading is 'Basic Settings' with the instruction 'Edit basic settings for creating database pool'. There is a Geronimo logo in the top right. Below the heading, there are three input fields: 'Pool Name' with 'jdbc/TestDataSource', 'Database Type' with a dropdown menu showing 'Derby Embedded XA', and 'Database Name' with 'TestDataBase'. Below these is a 'Driver Details' section with a text area containing the driver path 'org.apache.geronimo.configs/system-database//car'. At the bottom, there are navigation buttons: a help icon, '< Back', 'Next >', 'Cancel', and 'Finish'.

5. Click **Next** to go to the steps of **Advanced Pool Settings** and **Connection Pool Parameters**, and then fill in the required fields.

The following two figures show the fields in these two steps:

New Database Pool

Advanced Pool Settings
Edit advanced settings for creating database pool

Advanced Settings

☒ Create Database

Login Timeout: 5000

Login Details

User ID: system

Password: *****

Confirm password: *****

< Back Next > Cancel Finish

New Database Pool

Connection Pool Parameters
Edit Connection Pool parameters for Database pool

Connection Pool Parameters

Transaction Type: XA

Min Size: 0

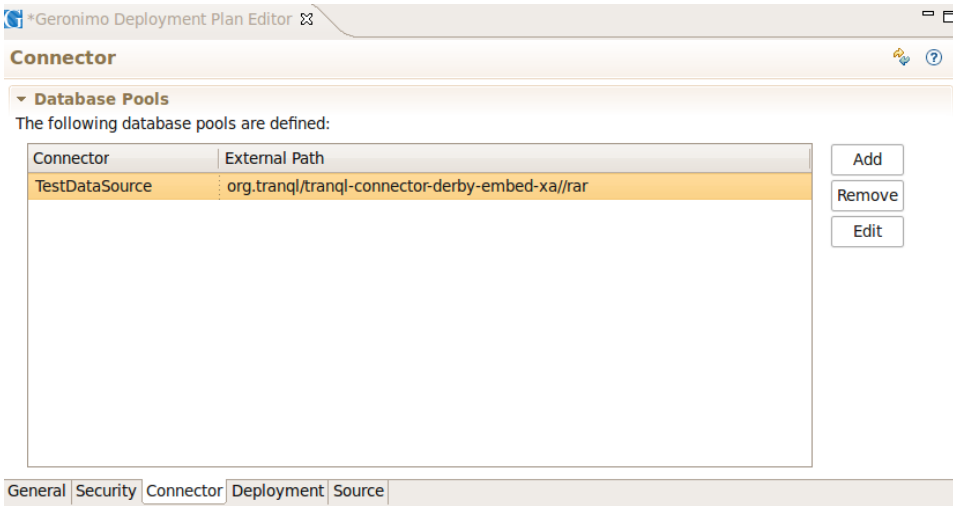
Max Size: 10

Blocking Timeout(in ms): 5000

Idle Timeout(in mins): 30

< Back Next > Cancel Finish

6. Click **Finish** to create the new datasource pool. The following figure shows the **Connector** tab with the new datasource pool that is created successfully:



7. In the toolbar, click **Save** to save the settings of the new datasource pool.

Now you have created a new datasource pool. You can click the **Source** tab to review the deployment plan of the connector. The following figure shows a sample deployment plan with a new datasource connector embedded:

geronimo-application.xml

```
<?xml version="1.0" encoding="UTF-8"?><app:application application-name="TestEAR" xmlns:app="http://geronimo.apache.org/xml/ns/j2ee/application-2.0"
xmlns:client="http://geronimo.apache.org/xml/ns/j2ee/application-client-2.0" xmlns:conn="http://geronimo.apache.org/xml/ns/j2ee/connector-1.2"
xmlns:dep="http://geronimo.apache.org/xml/ns/deployment-1.2" xmlns:ejb="http://openejb.apache.org/xml/ns/openejb-jar-2.2"
xmlns:log="http://geronimo.apache.org/xml/ns/loginconfig-2.0" xmlns:name="http://geronimo.apache.org/xml/ns/naming-1.2"
xmlns:pers="http://java.sun.com/xml/ns/persistence" xmlns:pkgen="http://openejb.apache.org/xml/ns/pkgen-2.1"
xmlns:sec="http://geronimo.apache.org/xml/ns/security-2.0" xmlns:web="http://geronimo.apache.org/xml/ns/j2ee/web-2.0.1">
  <dep:environment>
    <dep:moduleId>
      <dep:groupId>default</dep:groupId>
      <dep:artifactId>TestEAR</dep:artifactId>
      <dep:version>1.0</dep:version>
      <dep:type>car</dep:type>
    </dep:moduleId>
    <dep:dependencies>
      <dep:dependency>
        <dep:groupId>org.apache.geronimo.configs</dep:groupId>
        <dep:artifactId>system-database</dep:artifactId>
        <dep:type>car</dep:type>
      </dep:dependency>
      <dep:dependency>
        <dep:groupId>org.apache.geronimo.framework</dep:groupId>
        <dep:artifactId>j2ee-security</dep:artifactId>
        <dep:type>car</dep:type>
      </dep:dependency>
    </dep:dependencies>
  </dep:environment>
  <app:ext-module>
    <app:connector>TestDataSource</app:connector>
    <app:external-path>
      <dep:groupId>org.tranql</dep:groupId>
      <dep:artifactId>tranql-connector-derby-embed-xa</dep:artifactId>
      <dep:type>rar</dep:type>
    </app:external-path>
    <conn:connector>
      <conn:resourceadapter>
        <conn:outbound-resourceadapter>
          <conn:connection-definition>
```

```

interface>
    <conn:connectionfactory-interface>javax.sql.DataSource</conn:connectionfactory-
    <conn:connectiondefinition-instance>
        <conn:name>jdbc/TestDataSource</conn:name>
        <conn:config-property-setting name="Password">manager</conn:config-property-setting>
        <conn:config-property-setting name="LoginTimeout">5000</conn:config-property-
setting>
        <conn:config-property-setting name="DatabaseName">TestDataBase</conn:config-
property-setting>
        <conn:config-property-setting name="CreateDatabase">true</conn:config-property-
setting>
        <conn:config-property-setting name="UserName">system</conn:config-property-setting>
        <conn:connectionmanager>
            <conn:xa-transaction>
                <conn:transaction-caching/>
            </conn:xa-transaction>
            <conn:single-pool>
                <conn:max-size>10</conn:max-size>
                <conn:min-size>0</conn:min-size>
                <conn:blocking-timeout-milliseconds>5000</conn:blocking-timeout-
milliseconds>
                <conn:idle-timeout-minutes>30</conn:idle-timeout-minutes>
                <conn:match-one/>
            </conn:single-pool>
        </conn:connectionmanager>
    </conn:connectiondefinition-instance>
</conn:connection-definition>
</conn:outbound-resourceadapter>
</conn:resourceadapter>
</conn:connector>
</app:ext-module>
</app:application>

```