

Quick Start Guide

Using IBM DB2 With Mural Master Index Studio

Patch V1.0

@2009 NetGen Software, Inc.

Parts of the patch are derived from Mural Open Master Data Management Community where source codes and materials are under CDDL 1.0. Parts that the patch interacts with and are used in the guide are free download and open sourced from their respective communities and companies.

Preface

Mural Master Index Studio now supports IBM DB2 9 database server. The patch that supports DB2 works with Sun Master Data Manager (MDM) suite and Sun Enterprise Service Bus (ESB) suite. There are a few differences in creating the master index database tables and defining the database connection when using DB2. This guide describes basic steps for using and configuring master index to utilize IBM DB2 9.

Who Should Use This Guide

This guide is intended for use by developers who create, assemble and deploy master index composite applications and are need to interact with IBM DB2 database servers.

Quick Facts

Database Support

- IBM DB2 9.5 Enterprise Server Edition
- IBM DB2 9.5 Express Edition
- IBM DB2 9.5 Express-C Edition
- IBM DB2 UDB v 8.1 and v 8.2

According to IBM DB2 web site, DB2 v8.2 or v8.1 is going out of base support. IBM DB2 9.5 is suggested version and IBM DB2 9.5.2 Express-C Edition is used for development and testing while the patch is announced.

Sun MDM and ESB Support

- Sun MDM v6 Release
- Glassfish ESB v2 Release
- Sun Java Composite Platform Suite v6 Release

Where To Get

- [IBM DB2 9.5.2 Express-C Edition](#)

<http://www-01.ibm.com/software/data/db2/express/download.html>

- [Sun Java CAPS](#)

<http://www.sun.com/software/javaenterprisesystem/javacaps/index.jsp>

- [Sun MDM suite](#)

<https://mural.dev.java.net/Downloads.html>

- [Glassfish ESB v2](#)

<https://open-esb.dev.java.net/Downloads.html>

Mural Master Index DB2 Patch v 1.0

db2patch-1.0.jar

Prerequisites

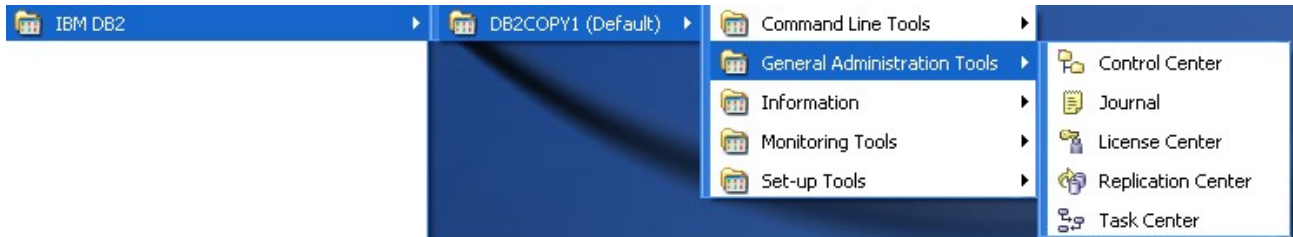
Before you start, you choose one of supported components, Sun Java CAPS 6, Glassfish ESB v2 or Sun MDM suite 6. and install it under your work machine. This guide assumes you work on windows XP/Vista machine and install Java CAPS 6 under C:\JavaCAPS6.

Installing Mural Master Index DB2 Support Patch

- `java -jar db2patch-1.0.jar [glassfish_home] [netbeans_home]`
For example, `java -jar db2patch-1.0.jar C:\JCAPS6DB2\appserver C:\JCAPS6DB2\netbeans`

Installing DB2 9 Database Server

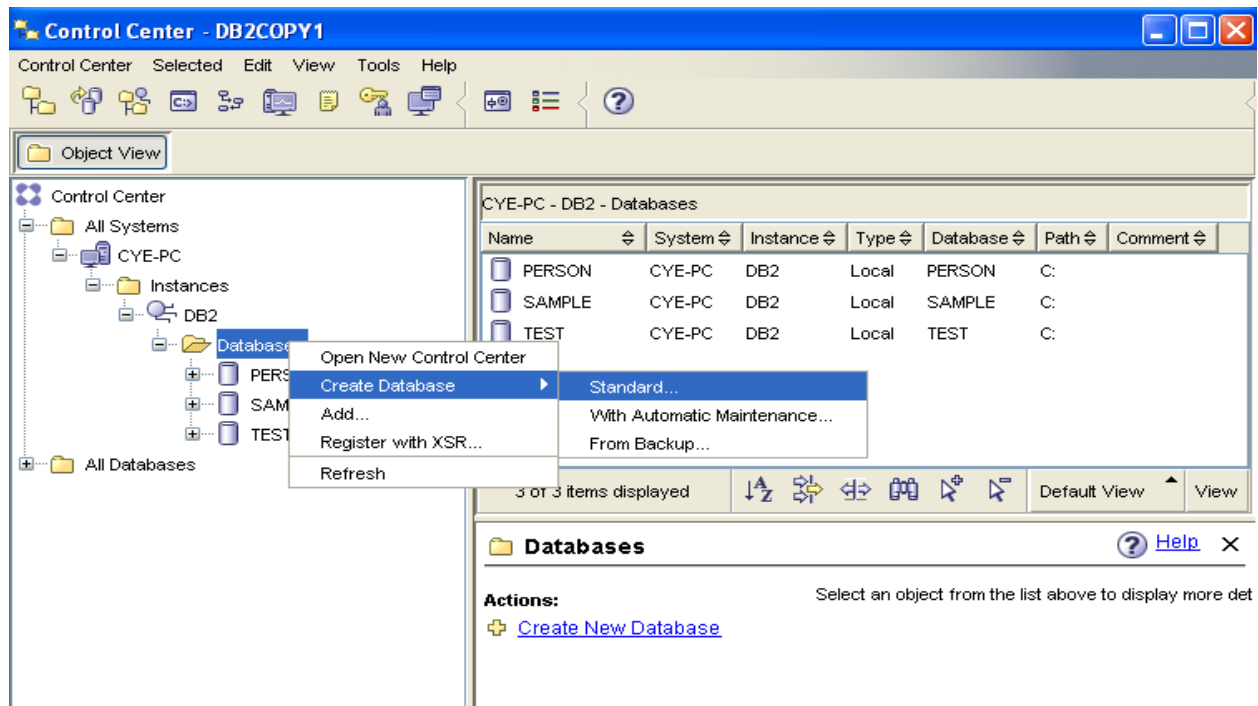
- Unzip the downloaded DB2 image.
- Run DB2 Setup wizard to install the product. The Setup installs DB2 database server, a set of DB2 tools including Command Line tools, Administration Tools, Monitoring Tools and more. Memorize DB2 basic settings, e.g., hostname:localhost; port:50000; user id: root; password:adminadmin, etc.



- Launch Control Center to start or stop or monitor the database instances. A named DB2 database instance is created and underneath it a SAMPLE database is created by default.

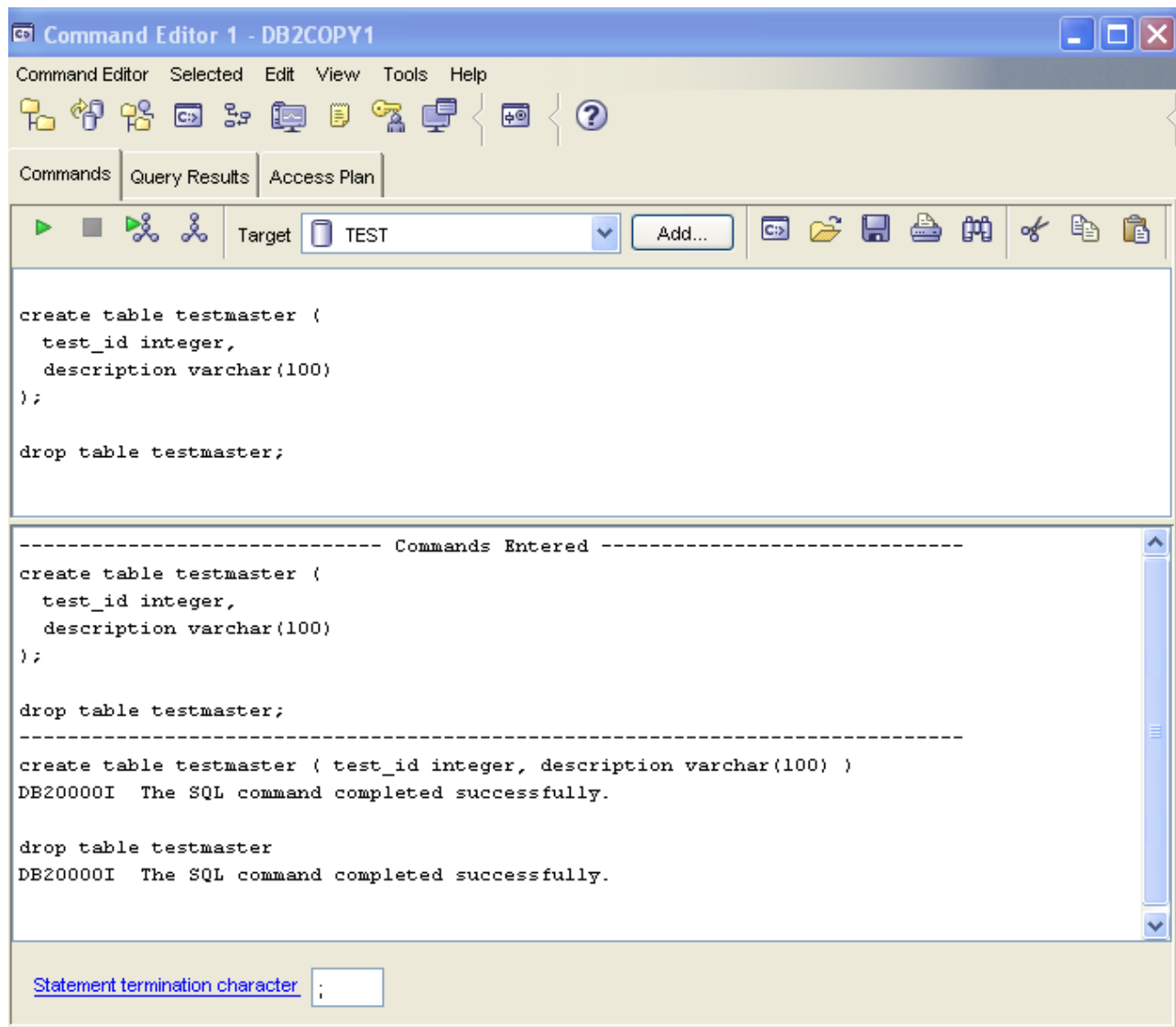
Creating Database

- Start up database instance using Control Center. e.g., you start DB2.
- Create a new database using Create Database Wizard launched from Control Center. e.g., you create a named PERSON database underneath DB2 instance.



Excise DB scripts.

- Launch Command Editor
- Set statement termination character as “;” from Tools Settings dialog.
- Add Target database.
- Copy and paste SQL statements in Commands window. Or you manually enter SQL scripts. e.g.,
create table testmaster {
test_id integer,
test_description varchar(100)
};
drop table testmaster;
- Click the green execution button to execute SQL scripts you entered in Commands window.



Adding the DB2 Driver to the Application Server

- Copy db2jcc.jar, db2jcc4.jar and db2jcc_license_cu.jar from db2_server_home\SQLLIB\java into app_server_home\lib if these jars are not under app_server_home\lib.
- Start the application server.
- Login the Admin Console.
- Add DB2 shared native libraries path into the application server path settings.
- Restart the application server.

Sun Java™ System Application Server Admin Console

The screenshot shows the Sun Java System Application Server Admin Console. On the left is a tree view with 'Application Server' selected. The main area has three sections:

- Classpath Prefix:** A text box with the label 'Classpath required for server operation (read-only)'.
- Classpath Suffix:** A text box with the label 'Prepend to server classpath' containing the value 'C:\JavaCAPS6\appserver\addons\accessmanager\am\WebServicesProvider.jar'.
- Native Library Path Prefix:** A text box with the label 'Append to server classpath' containing the value 'C:\DB2\IBM\SQLLIB\BIN'.

You can add native shared libraries path into JVM options. Either ways are fine.

The screenshot shows the Sun Java System Application Server Admin Console with the 'JVM Options' page selected. The page has a 'Save' button and a table of options.

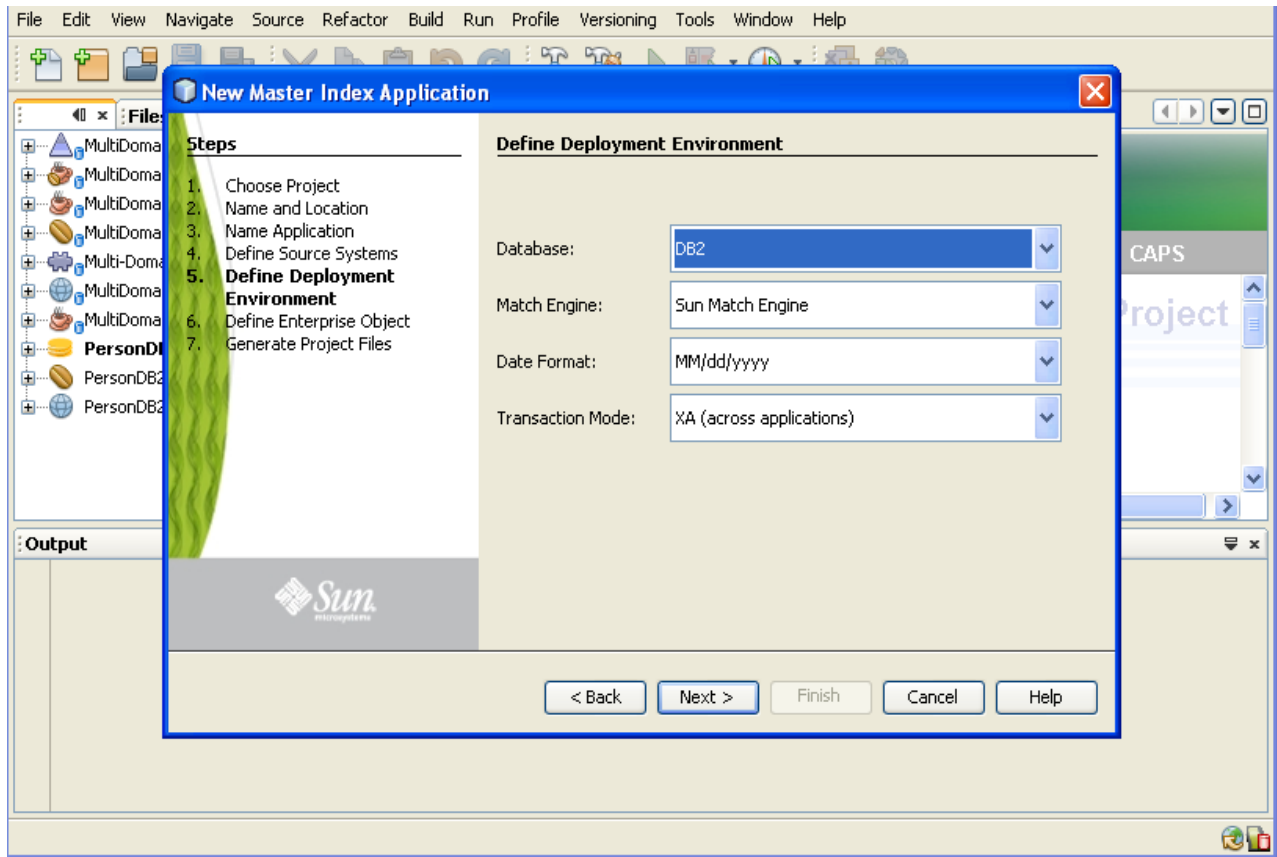
JVM Options

Manage JVM options for the server. An option value with a space needs to have enclosing double quotes. Click the Save button to save any changes.

Options (19)	Value
<input type="checkbox"/>	-Djava.library.path="C:\DB2\IBM\SQLLIB\BIN"
<input type="checkbox"/>	-XX:MaxPermSize=192m
<input type="checkbox"/>	-Djava.endorsed.dirs=\${com.sun.aas.installRoot}/lib/endorsed
<input type="checkbox"/>	-Djava.security.policy=\${com.sun.aas.instanceRoot}/config/server.policy

Creating Master Index Application

- Create Master Index Application using NetBeans as usual, choose DB2 database type during creating new Master Index Application project. e.g., PersonDB2.
- Build Master Index Application you create.



Defining the Database Connection Pools

- Log in to the Application Server Admin Console.
- In the left portion of the Admin Console, expand Resources, expand JDBC, and select Connection Pools.
- On the Create Connection Pool page, click New.
- In the Name field, enter a name for the connection pool. e.g., DB2ConnectionPoolForPersonDB2.
- In the Resource Type field, select the Java class for the type of transactions the master index application processes. `javax.sql.DataSource` or `javax.sql.ConnectionPoolDataSource` is for local transactions; `javax.sql.XADataSource` is for distributed transactions. e.g., select `javax.sql.XADataSource`.
- In the Database Vendor field, select DB2
- Click Next.
- In the DataSource Classname field, enter the Java class for the data source, or change the default value if it is provided to DB2 DataSource class. e.g., `com.ibm.db2.jcc.DB2XADataSource`.
- Modify the Pool Settings, Connection Validation, and Transaction properties according to your business practices. All defaults can be used for the sample project.
- In the additional properties section, enter values for the following properties:

url: The URL that points to the database. The syntax of the URL is:

`jdbc:db2://server:port/database_name.`

user: The login ID for the user you created for the master index database.

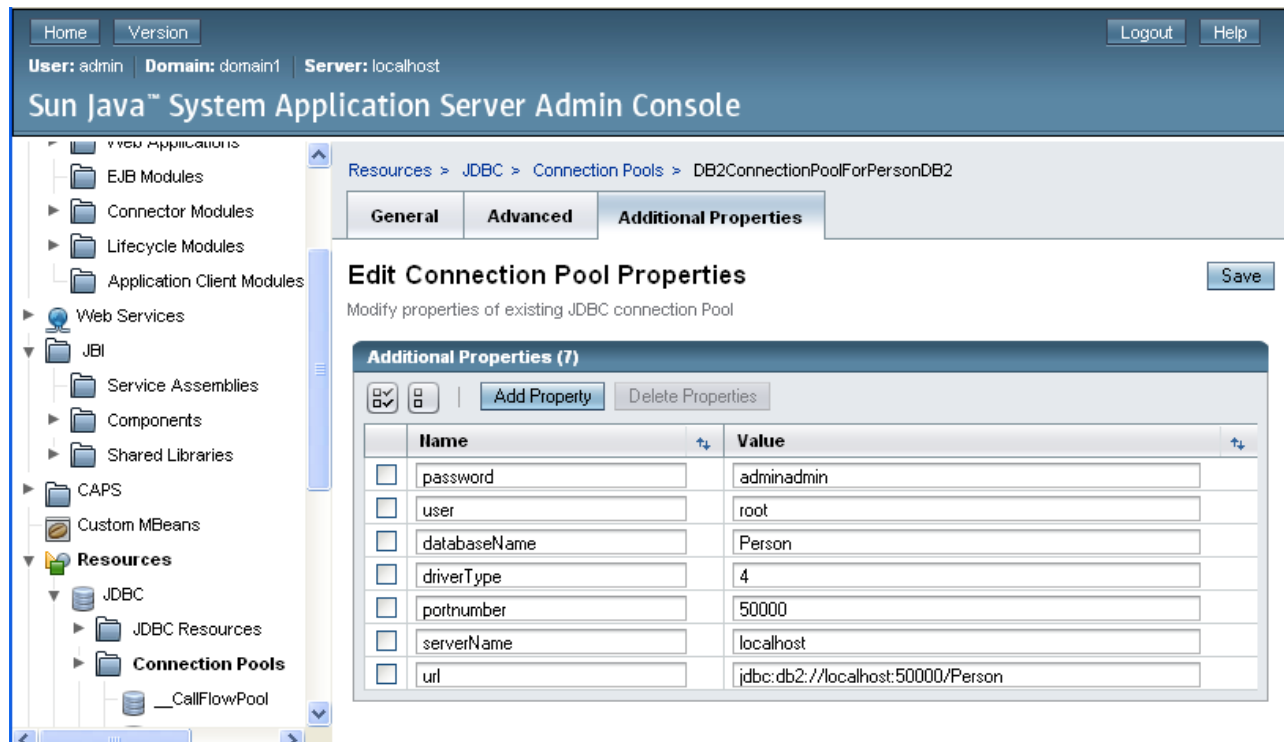
password: The password for the above user.

databaseName: The name of the DB2 database.

driveType: DB2 Universal JDBC driver type. Type 2 and Type 4 are supported. Only type 4 is tested while the patch is announced.

serverName: the host name which DB2 server runs.

portNumber: the listen port of DB2 server.



Creating JDBC Resources

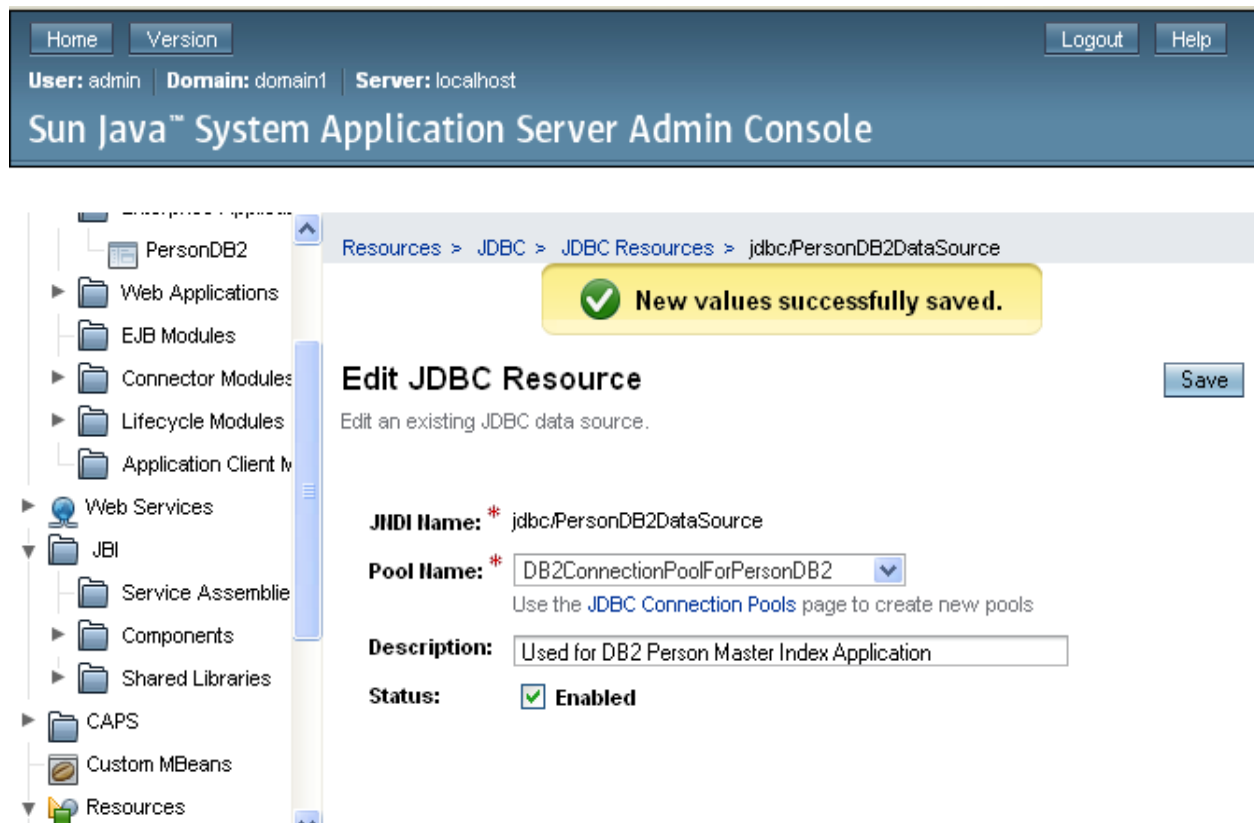
- In the left portion of the Admin Console, expand Resources, expand JDBC, and select JDBC Resources.
- On the Create JDBC Resource page, click New.
- In the JNDI Name field, enter a unique name for the JDBC resource in the form `jdbc/application_nameDataSource`, where *application_name* is the name of the master index application.

For example, `jdbc/PersonDB2DataSource`.

- In the Pool Name field, enter the name of the JDBC connection pool you created earlier. For example, `DB2ConnectionPoolForPersonDB2`.
- In the Status field, select the Enabled check box.
- Click OK.
- Repeat the previous steps to create a JDBC resource for the sequence manager with these guidelines:

In the Pool Name field, enter the name of the second JDBC connection pool your created.

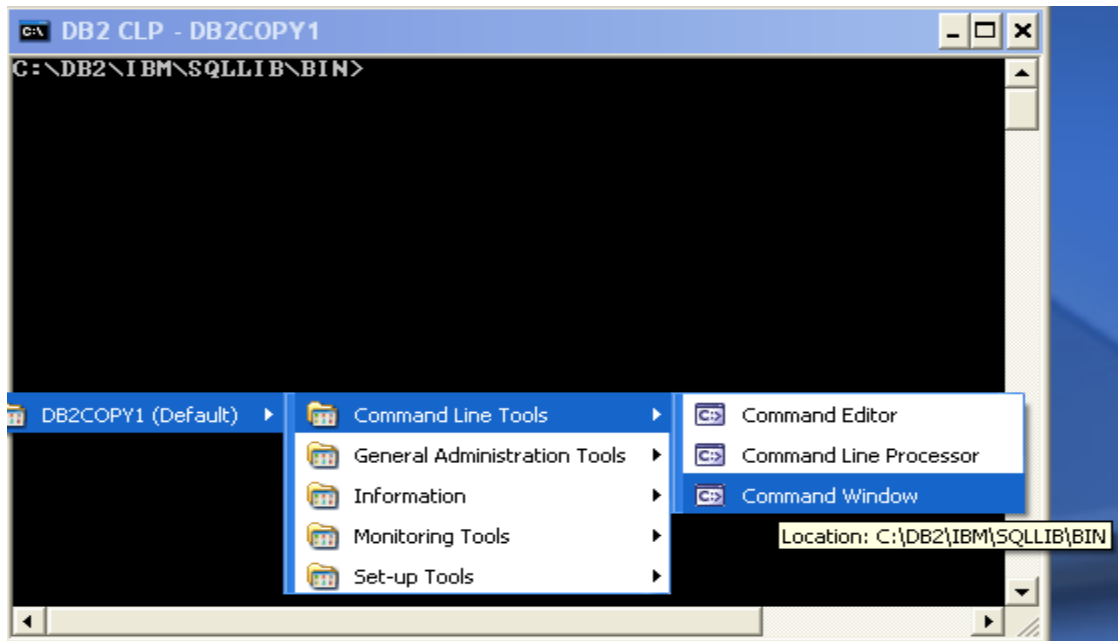
The name of the JDBC resource must be in the format `jdbc/application_nameSequenceDataSource`, where *application_name* is the name of the master index application. For example, `jdbc/PersonDB2SequenceDataSource`.



Creating Master Index Tables

Before you deploy the Master Index Application you create to use DB2 database server, for example, PersonDB2, you need create Master Index Tables. The database scripts, `create.sql`, `systems.sql`, `codelist.sql` and `drop.sql` can be found under `project_home\src\DatabaseScript`.

- Start Command Window



- Connect to the database by typing command:
db2_home\BIN\db2 connect to [database_name] user [user_name] using [password]
For example, connect to Person database you create early, you do:
db2_home\BIN\db2 connect to Person user root using adminadmin
- Execute master index scripts to create tables:
db2_home\BIN\db2 -td@ -f project_home\src\DatabaseScript\create.sql
db2_home\BIN\db2 -td@ -f project_home\src\DatabaseScript\systems.sql
db2_home\BIN\db2 -td@ -f project_home\src\DatabaseScript\codelist.sql
- To drop all tables if need, you can do:
db2_home\BIN\db2 -td@ -f project_home\src\DatabaseScript\drop.sql
- Disconnect to the database once you finish creating master index tables, do
db2_home\BIN\db2 terminate

Deploy Master Index Application

Now it is ready to deploy the project you create and enjoy all functions and features that master index application provides.

