Documentation DBCopy Eclipse Plugin

$Revision: 1.4 $ [$Date: 2003/08/10 14:32:16 $]

by Oliver Gries

Warning:

We are sorry, but the requested document is not available for your choosen language (en). The version provided is the default language version (de).

1. Documentation

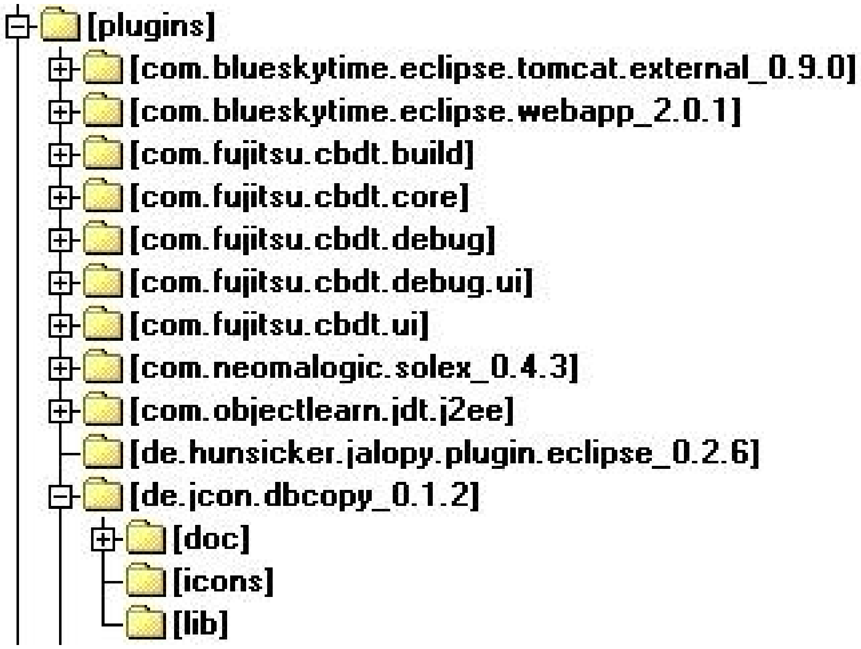
There are three simple steps to get the plugin to work.

- 1. Install it
- 2. Configure it
- 3. Use it

1.1. Installation

The installation of this plugin is quite simple. Just unzip the content into the `$ECLIPSE_HOME/plugins` directory preserving the directory structure provided.

After unzipped the directory structure should look like:



```
[plugins]
  [com.blueskytime.eclipse.tomcat.external_0.9.0]
  [com.blueskytime.eclipse.webapp_2.0.1]
  [com.fujitsu.cbdt.build]
  [com.fujitsu.cbdt.core]
  [com.fujitsu.cbdt.debug]
  [com.fujitsu.cbdt.debug.ui]
  [com.fujitsu.cbdt.ui]
  [com.neomalogic.solex_0.4.3]
  [com.objectlearn.jdt.j2ee]
  [de.hunsicker.jalopy.plugin.eclipse_0.2.6]
  [de.jcon.dbcopy_0.1.2]
    [doc]
    [icons]
    [lib]
```

Restart Eclipse to get the plugin activated.

1.2. Configuration

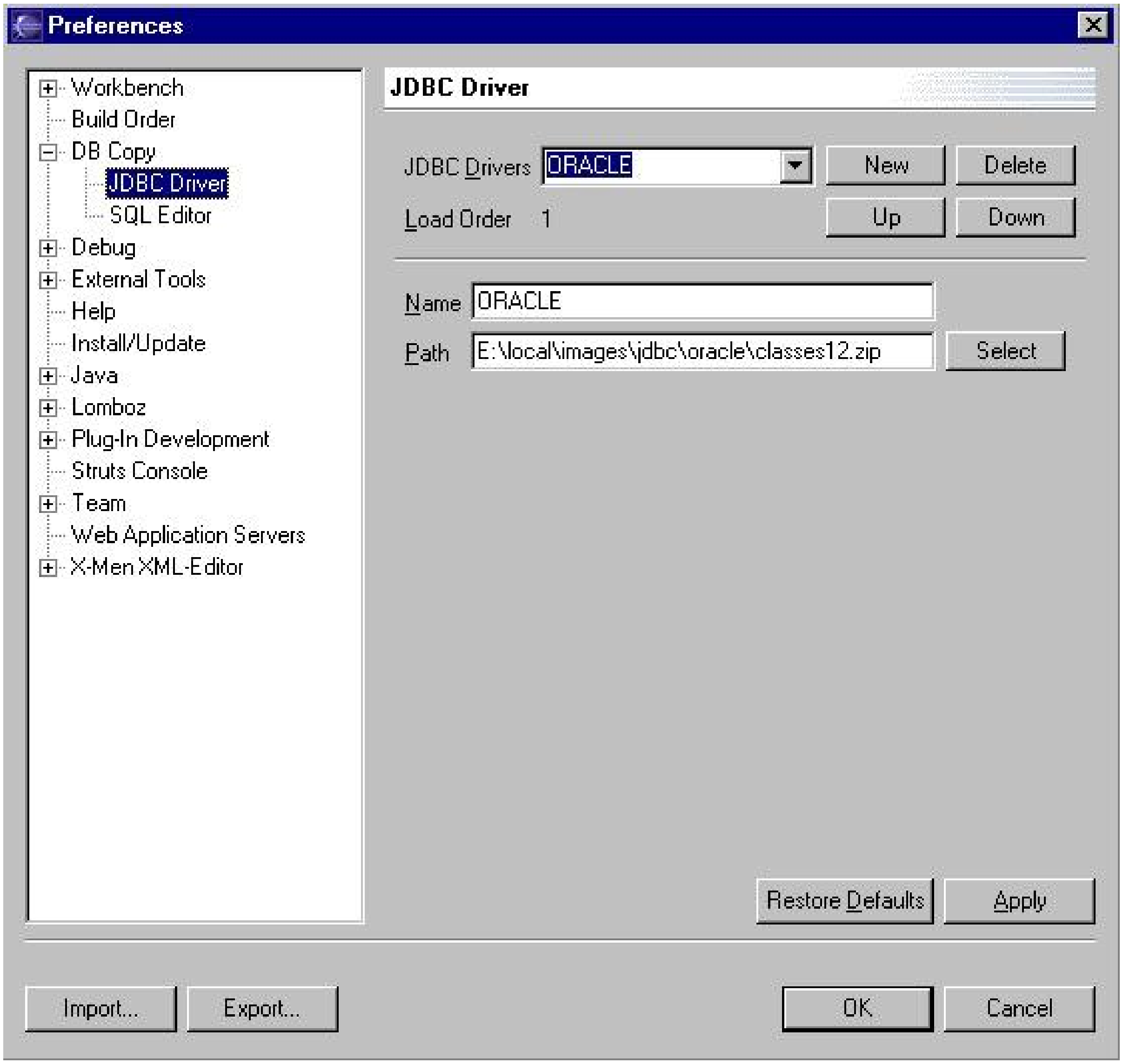
> **Note:**
>
> Verify the successfull installation by open the **Preferences** dialog of eclipse. An entry `DB Copy` should be provided in the list.

There are two steps needed to configure the plugin.

- [JDBC Driver] - Setup the JDBC driver and additional libraries
- [DB Locations] - Configure the connections

1.2.1. JDBC Driver Setup

Before you can use a driver you need to attach the library (jar or zip) of this driver to the classpath. This can be done dynamicly through the **JDBC Driver Preferences** page.



JDBC Driver

JDBC Drivers ORACLE

Load Order 1

Name ORACLE

Path E:\local\images\jdbc\oracle\classes12.zip

Specify a name and the location of the library.

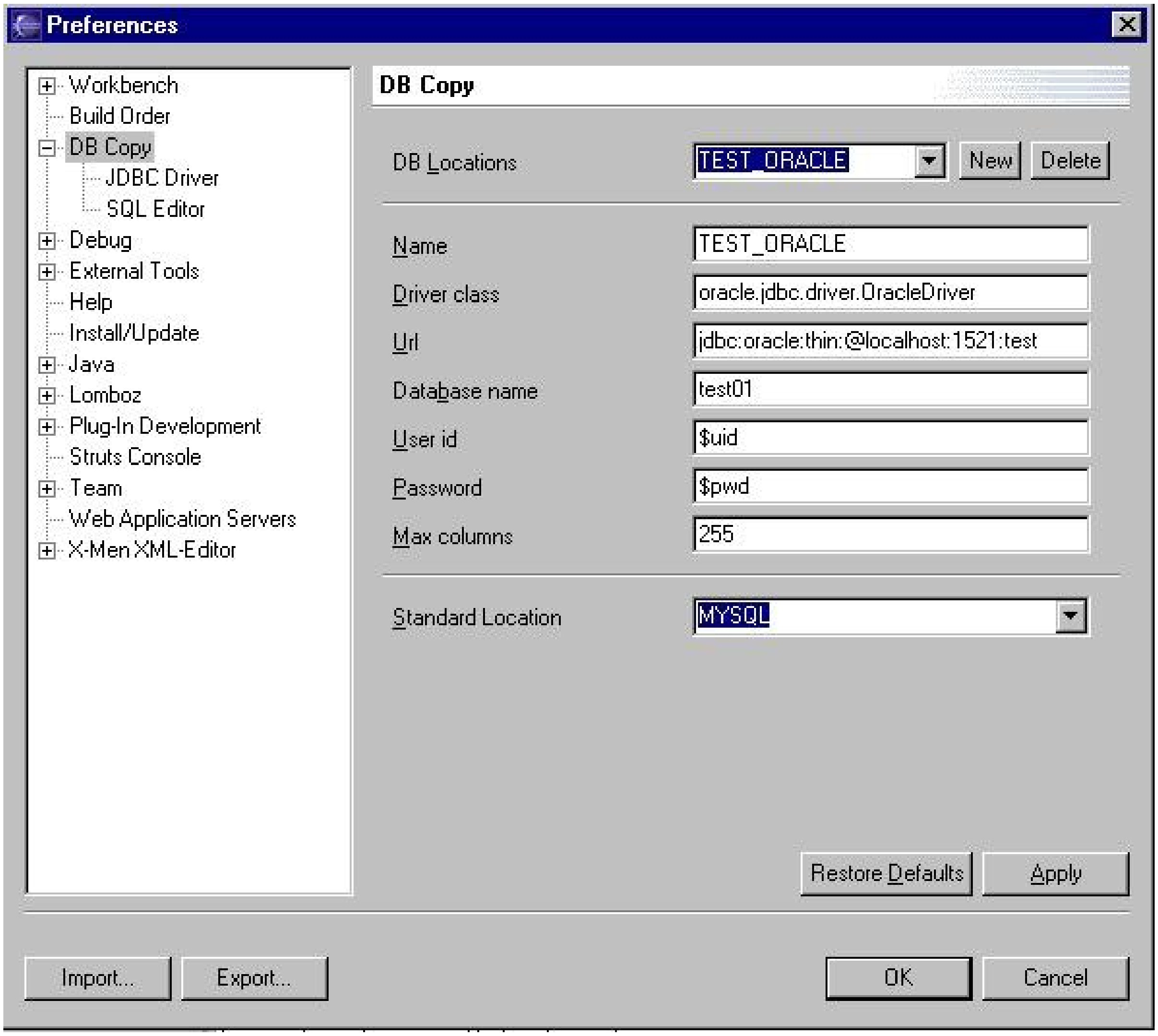
Note:

DB Copy doesn't test whether there is really a JDBC complient driver in this library. Infact this section isn't really just a JDBC Driver setup but an runtime extension of the classpath. You can attach any libraries needed. Some database provider do need additional libraries in the classpath as well to get the JDBC driver to work (e.g. QED needs the antlr.jar beeing placed in the classpath). With the free attachment of libraries this can be now achived quite simple.

With the **Load Order** buttons the attachment order to the class loader can be affected. This is importent because dependend libraries of JDBC driver need to be loaded before the driver itself. (e.g. The `antlr.jar` for QED needs to have a lower order position than the `qed.jar` driver itself)

1.2.2. DB Locations

Here you can configure your database connections. Each setup stands for one database location you can use to copy to/from.



DB Copy

DB Locations: TEST_ORACLE

Name: TEST_ORACLE

Driver class: oracle.jdbc.driver.OracleDriver

Url: jdbc:oracle:thin:@localhost:1521:test

Database name: test01

User id: $uid

Password: $pwd

Max columns: 255

Standard Location: MYSQL

Property	Description
Name	An unique name for this connection
Driver Class	The JDBC driver class e.g. `org.gjt.mm.mysql.Driver`
URL	The full JDBC URL to connect to the database. This is vendor specific so have a look at their documentation. For MySQL it might look like `jdbc:mysql://localhost:3306/test` whereby `test` is the database to connect to.
Database Name	Infact this is the alias which gets attached in front of any table in the SQL statement. If you don't need this you can leave this empty (from version 0.1.1 on, version 0.1.0 needs an entry here) and it's up to the JDBC driver to intepret it.
User ID	The userid for this connection
Password	The password for this connection
Max Columns	The max number of columns the driver supports. If you are unsure, use 255.
Standard Location	This connection gets first selected once you open the SQL Editor provided by this plugin.

1.2.3. SQL Editor

The only thing you can configure at the moment is the file extension DBCopy will use to create new SQL command files.

> **Note:**
>
> Version 0.1.0 had the extension `xsql` configured for DBCopy. From Version 0.1.1 on the extension is `dbcopy` which is associated as well with the SQL Command File Editor which comes with DBCopy. However it's your joice which extension you want to have, so feel free to configure it but please be aware that you might have to map your extension manually to DBCopy by adjusting the `File Extension Preferences` of Eclipse itself.

That's all. Of course there might be problems by using certain drivers, but this problems should be not different than normal developer praxis by using the JDBC driver in any application.

> **Note:**
>
> If you got problems to get the plugin to work, don't hesitate to contact us. But please, first have a look at the open bugs page and the latest version at our project home page `http://sourceforge.net/projects/dbcopy/` or

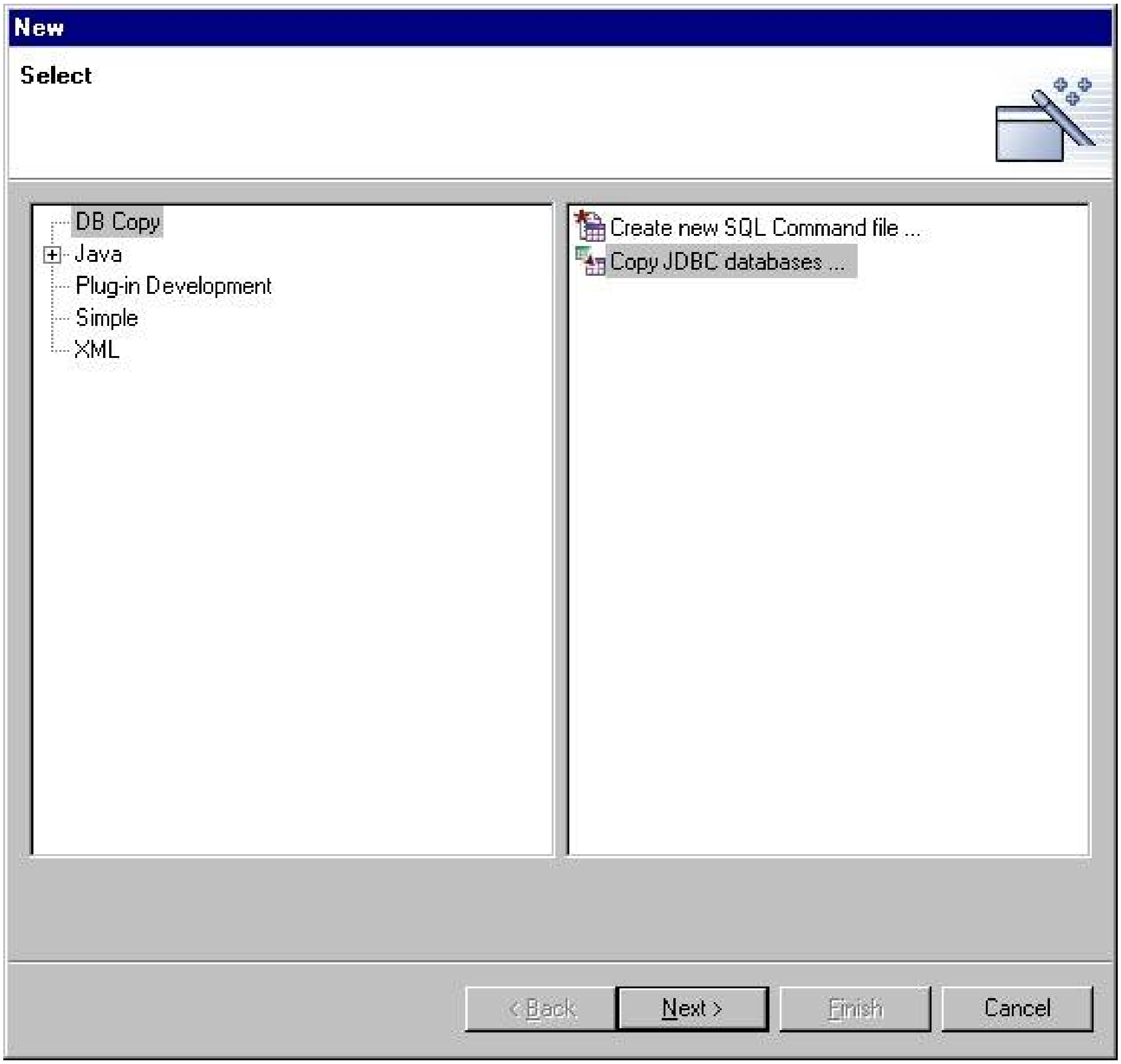
`y.sourceforge.net`.

1.3. Wizards, Editors and Views

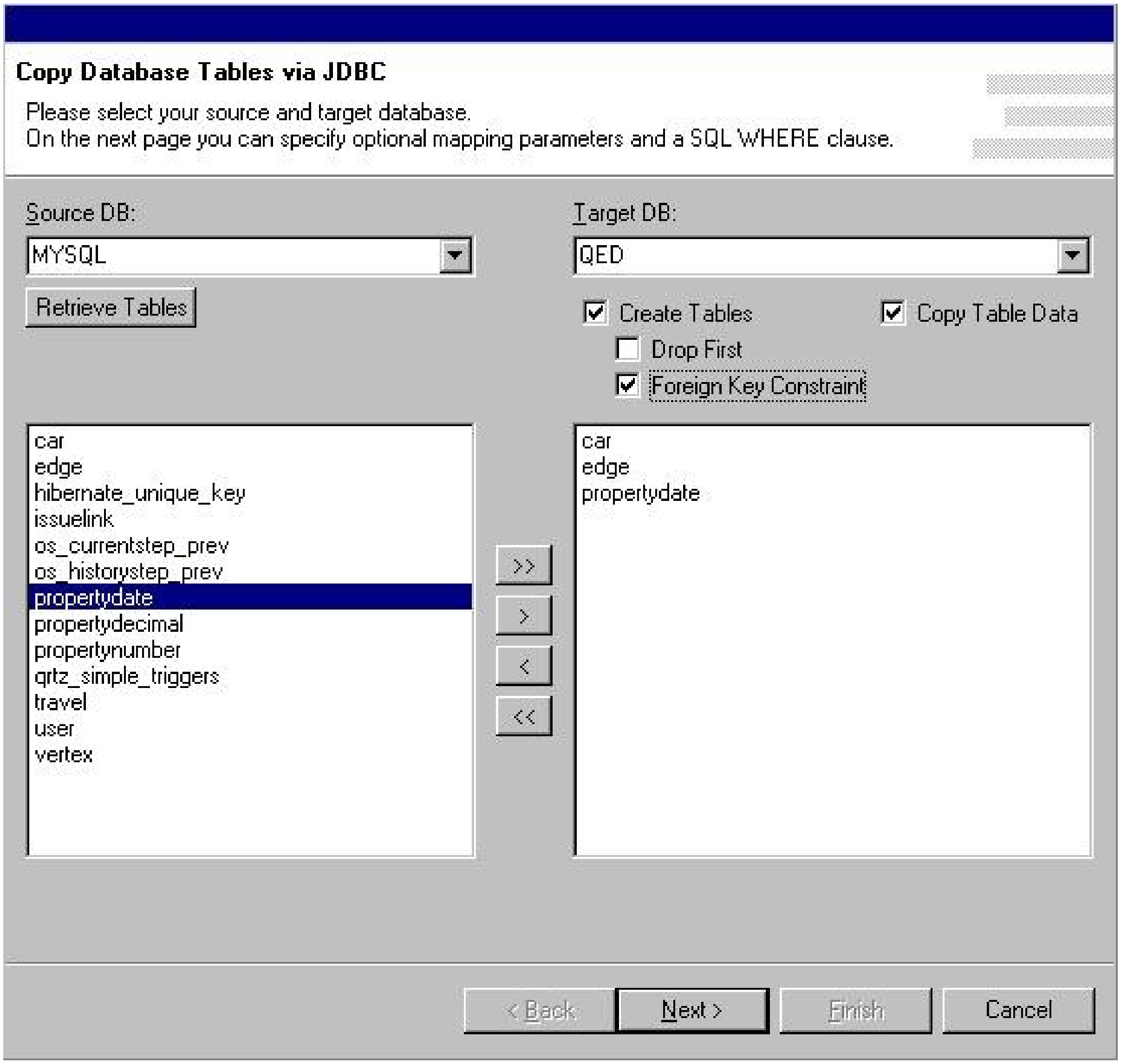
DB Copy provides following tools:

- A copy wizard
- A SQL editor and result view
- Ant tasks [from version 0.2.0 on]

1.3.1. DB Copy Wizard



Start this wizard thoughout the **New** action of eclipse (either [File - New - Other] or by selecting a project [New - Other]).



Copy Database Tables via JDBC

Please select your source and target database.
On the next page you can specify optional mapping parameters and a SQL WHERE clause.

Source DB: MYSQL

Target DB: QED

☑ Create Tables ☑ Copy Table Data

□ Drop First

☑ Foreign Key Constraint

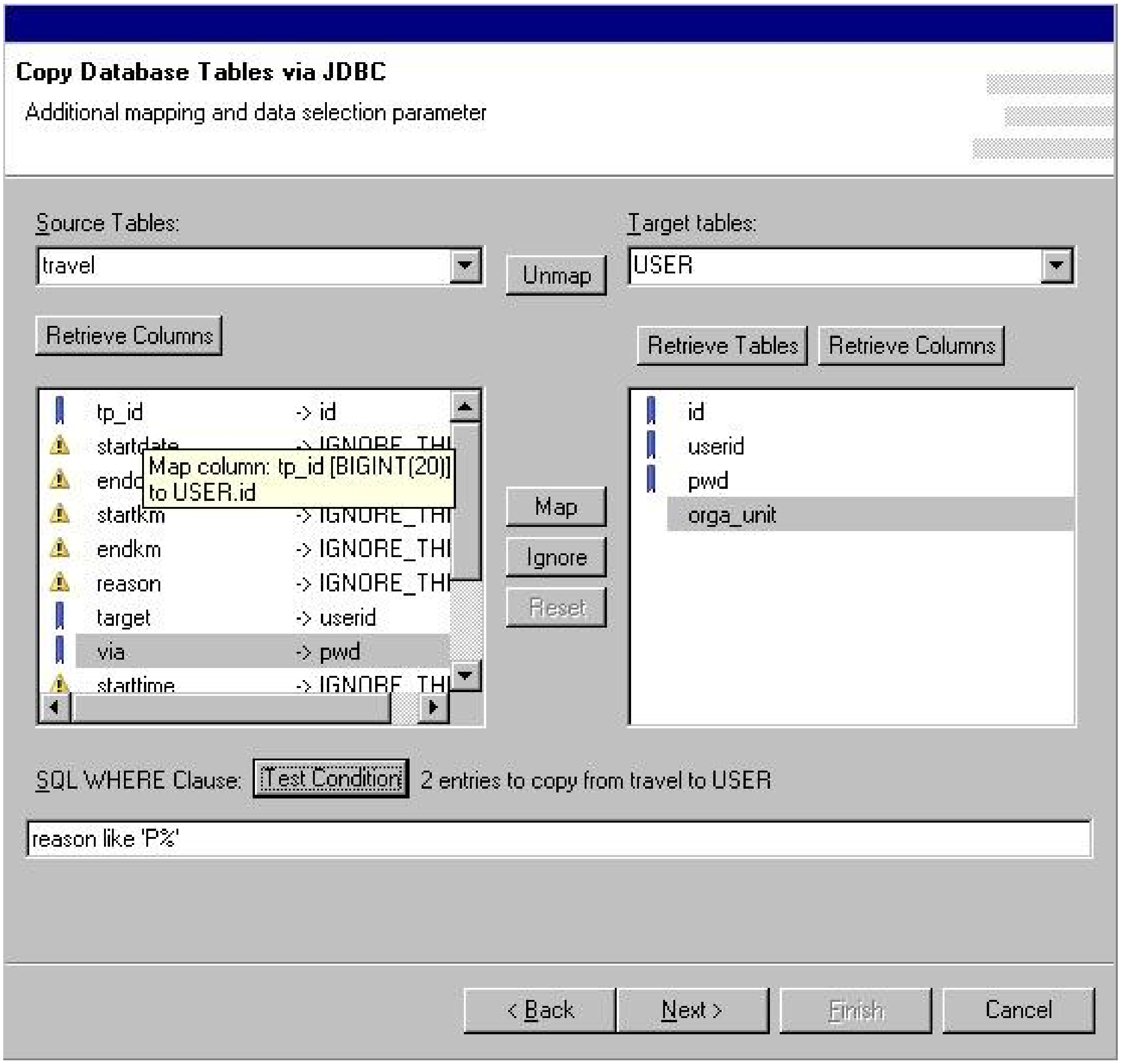
car
edge
hibernate_unique_key
issuelink
os_currentstep_prev
os_historystep_prev
propertydate
propertydecimal
propertynumber
qrtz_simple_triggers
travel
user
vertex

car
edge
propertydate

On the first page select the source and target db location (which you setup during the configuration process). Use **Retrieve Tables** to list the tables of the source db location. Choose the table(s) you want to copy. If you want the table to get automatically in your target db location created just switch the checkbox on. Press next if you want to proceed.

> **Note:**
> Of course you can use the source location dropdown list for having a look at the target db location tables as well.

Note that if you want to copy the Foreign Key Constraints as well the copy order of your tables matters.



Copy Database Tables via JDBC

Additional mapping and data selection parameter

Source Tables: travel

Target tables: USER

```
tp_id         -> id
startdate     -> IGNORE_THI
endd...
startkm       -> IGNORE_THI
endkm         -> IGNORE_THI
reason        -> IGNORE_THI
target        -> userid
via           -> pwd
starttime     -> IGNORE_THI
```

Map column: tp_id [BIGINT(20)] to USER.id

id
userid
pwd
orga_unit

SQL WHERE Clause: 2 entries to copy from travel to USER

reason like 'P%'

If you just want a one to one copy you will not need to provided additional mapping parameters. However with this option you can easily migrate data to other tables.

Saying you have an application which use a certain table structure. The new version of this application uses a different persistence mechanism to access non transient data. (EJB-CMP -> JDO, RO Mapping Tools, ...). Not every time it will be possible to preserve the old table structure. On this page is possible to map columns from the source table to columns of the target table. You can exclude columns to copy as well.

Note:

If we get feedback that this functionality is what you need or want to have, we fill focus on enhancing this feature (Converter for column mapping, the ability to provide standard values for non mapped columns, ...).

How to map?

You can only map if the `Create Tables` feature is switched off.

- Retrieve Target Tables
- Retrieve Columns for Source and Target Tables
- Map Columns to each other either by selecting the appropriate lines in the lists and hit the button between those lists or by simply using drag and drop. You can drag either from source to target or vice versa.
- Provide a `SQL WHERE` clause as needed.
- Hit the `Test Condition` button to see how many items will be copied.

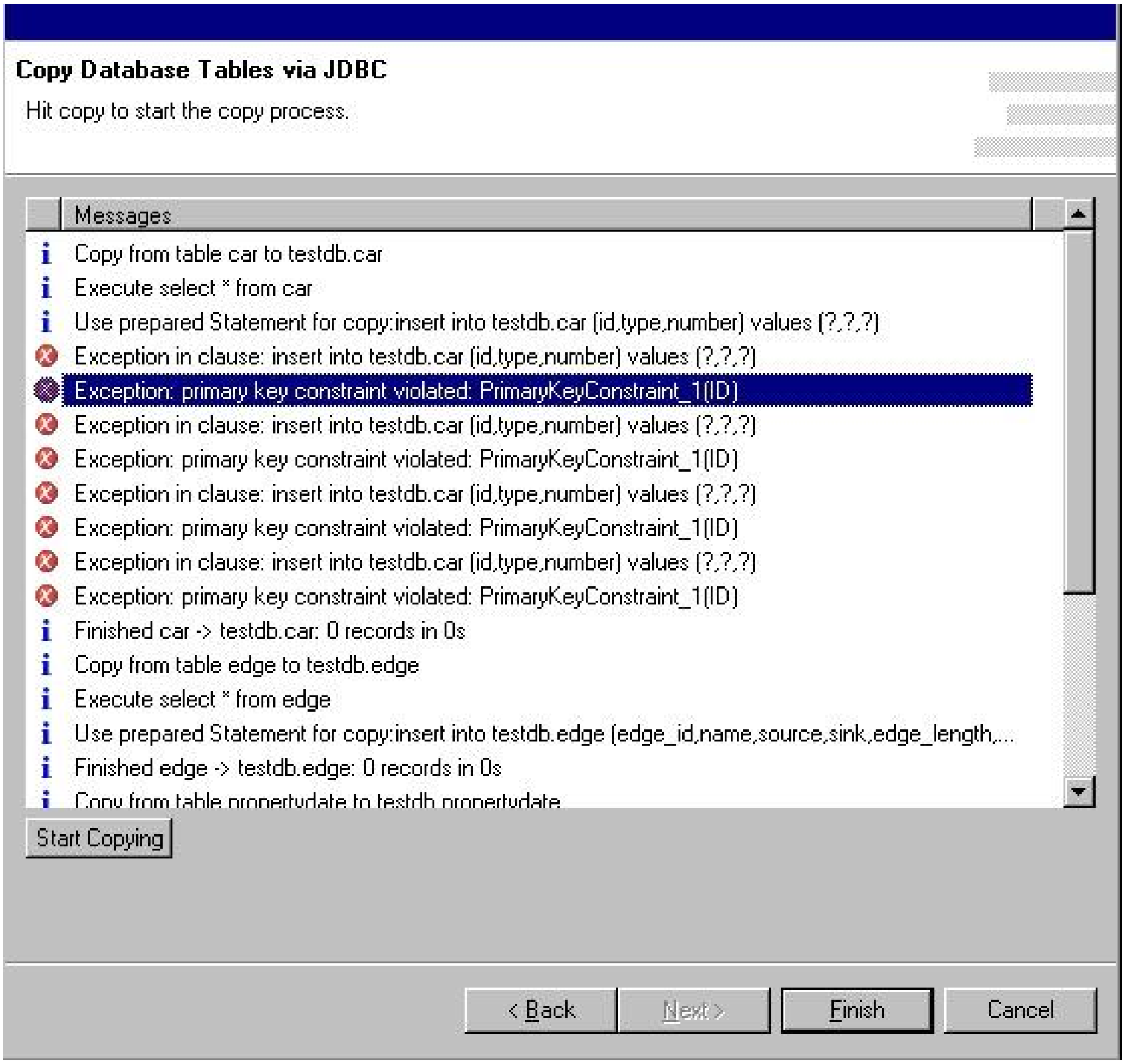
1.3.1.1. Standard Mapping Rules

The copy process is based on following rules:

Rule	Condition	Possible Problems	Actions
Standard Mapping	[Create Tables] is enabled. No column mapping can be specified.	Table allready exists	Copy process will not stop but try to proceed. If the existing table structure / name fits the source table, the copy process should be successfull
Standard Mapping	[Create Tables] is disabled. No column mapping has been specified.	Table doesn't exist. Table exists but doesn't fit the source table structure.	Copy process will not stop but try to proceed. It presumes that there are columns in the existing table which fit to the source table. The mapping is done by the equality of the table and column name. However if there are to many errors

			(Standard is 50, can be configured in the db.properties file of the plugin, in an later version within the preferences of th plugin itself), the copy data process will stop and proceed with the next table.
Standard Mapping	[Create Tables] is enabled. [Copy FK Constraints] is enabled. No column mapping can be specified.	The creation of FK constraints fails	The table will be created without FK constraints. The copy process will not stop. Note that tests with different databases have shown that this feature doesn't seem to be very stable. This is mainly due to different implementation of the IMPORTED KEYS specification of each JDBC driver. But give it a try you might be successfull anyway ;-)
Standard Mapping	[Drop Tables] is enabled.	Drop Tables fails	The copy process will not stop. Note that you can just drop all tables without copying any data by selecting only this feature and deselect all others (Don't know whether this is good at all ...).
Standard Mapping	[Create Tables] and [Drop Tables] is enabled.[FK Constraints]on your choice. [Copy Data] is disabled	see above	With this settings you can just recreate your table structure on a different database.
Additional Mapping	[Create Tables] is disabled. The source	If the column names differ, the copy will fail.	Stopping after the max number of errors has

	table is linked to a target table of your choice by hitting the [Map] button for tables. No additional column mappings has been specified.		been occured.
Additional Mapping	[Create Tables] is disabled. The source table is linked to a target table of your choice by hitting the [Map] button for tables. Additional column mappings has been specified.	If the target columns have a differnt type than the source columns, the copy will fail. If not all columns of the target table are mapped but are e.g. not nullable, the copy will fail. If not all columns of the source tables are mapped/ignored and there is no column with the same name on the target table, the copy will fail.	Stopping after the max number of errors has been occured.



Copy Database Tables via JDBC

Hit copy to start the copy process.

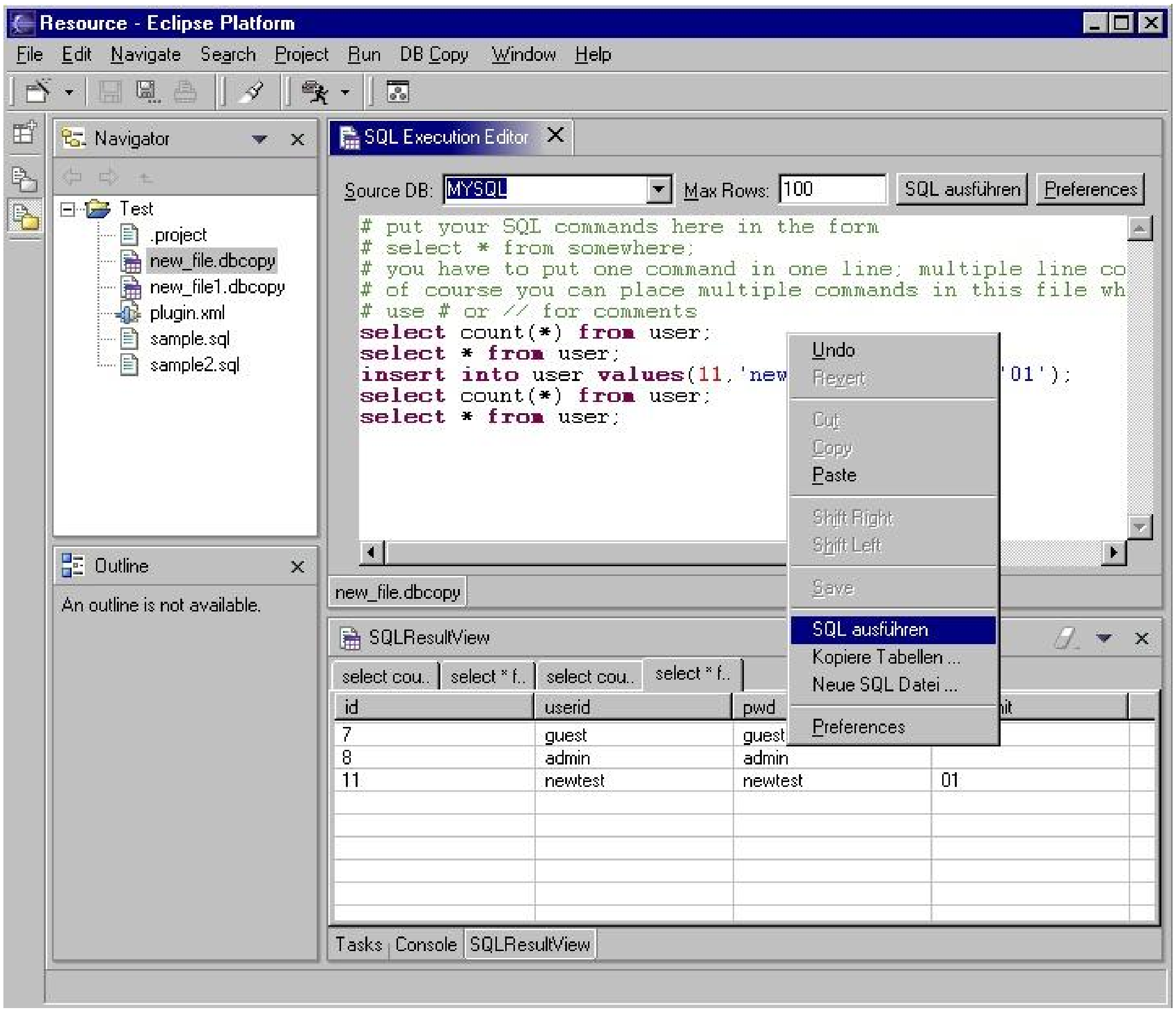
```
Messages
i  Copy from table car to testdb.car
i  Execute select * from car
i  Use prepared Statement for copy:insert into testdb.car (id,type,number) values (?,?,?)
⊗  Exception in clause: insert into testdb.car (id,type,number) values (?,?,?)
⊗  Exception: primary key constraint violated: PrimaryKeyConstraint_1(ID)
⊗  Exception in clause: insert into testdb.car (id,type,number) values (?,?,?)
⊗  Exception: primary key constraint violated: PrimaryKeyConstraint_1(ID)
⊗  Exception in clause: insert into testdb.car (id,type,number) values (?,?,?)
⊗  Exception: primary key constraint violated: PrimaryKeyConstraint_1(ID)
⊗  Exception in clause: insert into testdb.car (id,type,number) values (?,?,?)
⊗  Exception: primary key constraint violated: PrimaryKeyConstraint_1(ID)
i  Finished car -> testdb.car: 0 records in 0s
i  Copy from table edge to testdb.edge
i  Execute select * from edge
i  Use prepared Statement for copy:insert into testdb.edge (edge_id,name,source,sink,edge_length,...
i  Finished edge -> testdb.edge: 0 records in 0s
i  Copy from table propertydate to testdb.propertydate
```

Start the copy process on the 3rd page. A progress bar shows the current step.

> **Note:**
>
> If you want to copy a huge amount of data you should be patient, especially if you do it over a network connection.

After the copy process is finished you see the messages created during the copy action. Double click on one item to see the full message (especially good for exceptions).

1.3.2. SQL editor and result view



```
# put your SQL commands here in the form
# select * from somewhere;
# you have to put one command in one line; multiple line co
# of course you can place multiple commands in this file wh
# use # or // for comments
select count(*) from user;
select * from user;
insert into user values(11,'new                    '01');
select count(*) from user;
select * from user;
```

id	userid	pwd	it
7	guest	guest	
8	admin	admin	
11	newtest	newtest	01

The SQL editor provided by this plugin can execute multiple SQL statements. Select the DB Location you want to use and execute it. You can simple switch your DB Location and your SQL commands gets executed on a different database. The Result View gets automatically opened.

> **Note:**
>
> For version 0.1.x one SQL command needs to be in one row, it can't extend over multiple rows ;-(. This will work from version 0.2.0 on.

1.4. Additional Tasks

1.4.1. Configuration

You can change the standard file extension of the SQL Command File beeing created by the SQL Command File wizard in the preferences dialog as well. However you have to take into consideration that by changing the standard extension (which is `dbcopy` at the moment) you have to take care to register this extension to the apropriate editor on your own.

1.5. Common Problems

Have a look at the FAQ section.