

# Making BridgeDb Derby files with Groovy

Egon Willighagen 

Published September 9, 2023

## Citation

Willighagen, E. (2023). Making BridgeDb Derby files with Groovy. *Chem-bla-ics*. <https://doi.org/10.59350/pn744-knt64>

## Keywords

Groovy, Bridgedb

## Copyright

Copyright © Egon Willighagen 2023. Distributed under the terms of the [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

## chem-bla-ics

I just want to drop this here. There are various ways to make [BridgeDb](#) identifier mapping files. Some of the tools predate my joining the BiGCaT research group and the BridgeDb project, but this Groovy page is basically what we have been using to create the metabolite identifier mapping databases:

```
@Grab(group='org.bridgedb', module='org.bridgedb.bio', version='3.0.23')
@Grab(group='org.bridgedb', module='org.bridgedb.rdb.construct',
version='3.0.23')
```

```
import java.text.SimpleDateFormat;
import java.util.Date;
```

```
import org.bridgedb.IDMapperException;
import org.bridgedb.DataSource;
import org.bridgedb.Xref;
import org.bridgedb.bio.DataSourceTxt;
import org.bridgedb.rdb.construct.DBConnector;
import org.bridgedb.rdb.construct.DataDerby;
import org.bridgedb.rdb.construct.GdbConstruct;
import org.bridgedb.rdb.construct.GdbConstructImpl4;
```

```
DataSourceTxt.init()
```

```
GdbConstruct database = GdbConstructImpl4.createInstance(
    "test", new DataDerby(), DBConnector.PROP_RECREATE
);
database.createGdbTables();
database.preInsert();
```

```
inchikeyDS = DataSource.getExistingBySystemCode("Ik")
lmDS = DataSource.getExistingBySystemCode("Lm")
swisslipidsDS = DataSource.getExistingBySystemCode("Sl")
```

```
String dateStr = new SimpleDateFormat("yyyyMMdd").format(new Date());
database.setInfo("BUILDDATE", dateStr);
database.setInfo("DATASOURCENAME", "LIPIDMAPS_SWISSLIPIDS");
database.setInfo("DATASOURCEVERSION", "LIPID_TEST");
database.setInfo("DATATYPE", "Metabolite");
database.setInfo("SERIES", "standard_metabolite");
```

```
ref1 = new Xref("YECLLIMZHNYFCK-RRNJGNTNSA-J", inchikeyDS, true);
ref2 = new Xref("LMFA07050035", lmDS, false);
database.addGene(ref1)
```

## chem-bla-ics

```
database.addGene(ref2)
database.addLink(ref1, ref1)
database.addLink(ref1, ref2)

ref3 = new Xref("SLM:000000493", swisslipidsDS, true);
database.addGene(ref3)
database.addLink(ref1, ref3)

database.commit();
database.finalize();
```

For the people who have worked with BridgeDb Java in the past, note the new SQL schema 4, as used by the `GdbConstructImpl4`. This schema allows indicating of an identifier is outdated/retired/etc. This is actually the case for the `LMFA07050035` identifiers, and hence the `false` parameter in the `new Xref()` call.